(2)

System Study: Final Report
Radiation Hardened Autonomous Navigation

January 19, 1989

Prepared For:
Naval Research Laboratory
Contract N00014-85-C-2536

Charlie Rose
Principal Math. Analyst
Principal Investigator

Dr. Frank Wargocki
Principal E-O Engineer

Pamela Stewart
Acting Dept. Manager
Electro-Optics Tech. Devel.

C.L. Comer
Contracts Manager
Electro-Optics Subsystems

Don Vanlandingham
Vice President
Electro-Optics Subsystems

Ball Aerospace Systems Group
P.O. Box 1062
Boulder, Colorado

Contents

List of Figures

List of Tables

1 Introduction

The knowledge of a spacecraft's position in it's orbit has always been of paramount importance, and the estimation of the orbit has historically been a ground-based operation. The computation facilities required to handle the sophisticated optimal estimation algorithms have been of the VAX class, or larger, and, of necessity, been confined to well-ventilated computer centers. This is changing. CPUs which can outperform a VAX 11/780 by factors of ten or twenty are now available in a two chip set which consume only a few watts. Orbit estimation, now being called Autonomous Navigation (AutoNav), can now be physically performed on its associated vehicle. This paper reports on our investigations of one such AutoNav scheme.

AutoNav systems for spacecraft comprise four major components:

Reference Points: Spacecraft orbit estimation requires knowledge of reference vectors to targets or bodies whose positions are known in inertial space. These measurements are sometimes used in conjunction with the knowledge of satellite attitude to determine the orbital parameters of the satellite. These known landmarks may be celestial bodies such as the sun, moon, or planets. They may be known points on the Earth or moon, or they may be other satellites in known orbits, such as GPS. Without compromising the utility of the general technique described, this study has considered the use of Earth-based landmarks to define a system that is easily realisable.

Sensor(s): Depending on the type of reference points used, sensors which observe these references may be ranging devices (e.g., when used with GPS or coöperative ground-based transmitters) or angle sensing devices. In this study, the latter type was used; an optical sensor of the NRL Rad-Hard Star Tracker class.

Measurement Processor: The measurements made by the sensor(s) on the reference point(s) must be converted into corrrections to the Au-

toNav system's opinion of where it is. The ideal algorithm for this is the Extended Kalman Filter.

Orbit Propagator: When the reference points are not visible to the AutoNav sensor, and measurements are not available, the AutoNav system must be able to project its position as time passes until new measurements come in. This is the job of the Orbit Propagator.

These four components are described much more fully in following sections. There are two approaches to landmark (Earth surface) tracking:

Imaging Sensor: This concept involves the use of an on-board imaging sensor, such as SPOT or other Earth imaging sensor, to extract the position of known topographic features. While such landmark data has been used to improve the attitude estimate of some systems, the use of this method has not been demonstrated in an operational attitude control system. It is assumed that high resolution can be achieved only after ground processing of the imager data. This limits the application of this type of data in a real-time autonomous navigation system.

Beacon/Target Sensor: Tracking with ground-based beacons involves using a point source tracker to acquire and track a coöperative target or beacon on the ground at a known location. The target could be a laser beacon source or a passive retroreflector used with an on-board illuminator. This concept can be extended to tracking beacons on other sources, such as the moon, aircraft, ships, or other satellites.

Sensor design, operation, and accuracy are essentially identical to those of a star tracker, allowing application of a well-developed technology. The beacon could employ a visible or near-infrared laser source, such as a laser diode, for use with a tracker employing a silicon charge transfer device (CTD) detector. The use of an active sensor illuminating a passive retroreflector on the ground has advantages in autonomy and reduced ground station complexity, but it requires that a relatively high powered laser be flown, adversely impacting the space segment of the system.

For this study, we have assumed tracking a set of fixed point sources on the Earth. For the purpose of estimating sensor performance and configuration,

active beacons are assumed, although, for the navigation simulation, this distinction is immaterial.

This, then, is essentially a feasibility study for doing Autonomous Navigation using a sensor of the NRL Rad-Hard Star Tracker class in an Earth-based landmark configuration.

2 AutoNav System: Description

In this section, the particular autonomous navigation system studied is described in detail. The software simulation used to test it is described in Sec. 3.

2.1 Overview

This study examines the behavior of an AutoNav system comprised of a single sensor very similar to the NRL Rad-Hard star tracker, an Extended Kalman Filter which processes the sensor measurements, and a fairly simple Keplerian orbit propagator. A block diagram of the system appears in Fig. 1 on page 6, and the operation of the system is described below.

1. Given an initial estimate, $\hat{\mathbf{u}}_0$, of the position and velocity of the vehicle, the orbit propagator is used to compute the position and velocity, $\hat{\mathbf{u}}(t)$, at subsequent times. This is a blind propagation, and is used during those intervals in which no reference points fall within the field of view of the sensor.

2. When one or more reference points *do* fall within the sensor's ken, the sensor output, $\mathbf{z}$, is given to the AutoNav system for processing.

3. The AutoNav system compares these sensor measurements to what it *thinks* the sensor outputs should be, $\hat{\mathbf{z}}$, based on its (probably erroneous) idea of where the vehicle is, $\hat{\mathbf{u}}(t)$. The Kalman Filter produces the means (the Kalman gain matrix, $\mathbf{K}$) by which these differences in measurements, Δz, can be converted into a correction, $\Delta \mathbf{u}$, of the vehicle's estimated position and velocity.

This process is repeated forever: fly blindly until measurements of known reference points permit refinements of the estimated position and velocity to be made. Continue making these refinements until the reference points *disappear from sight, and then go back to flying blindly.*

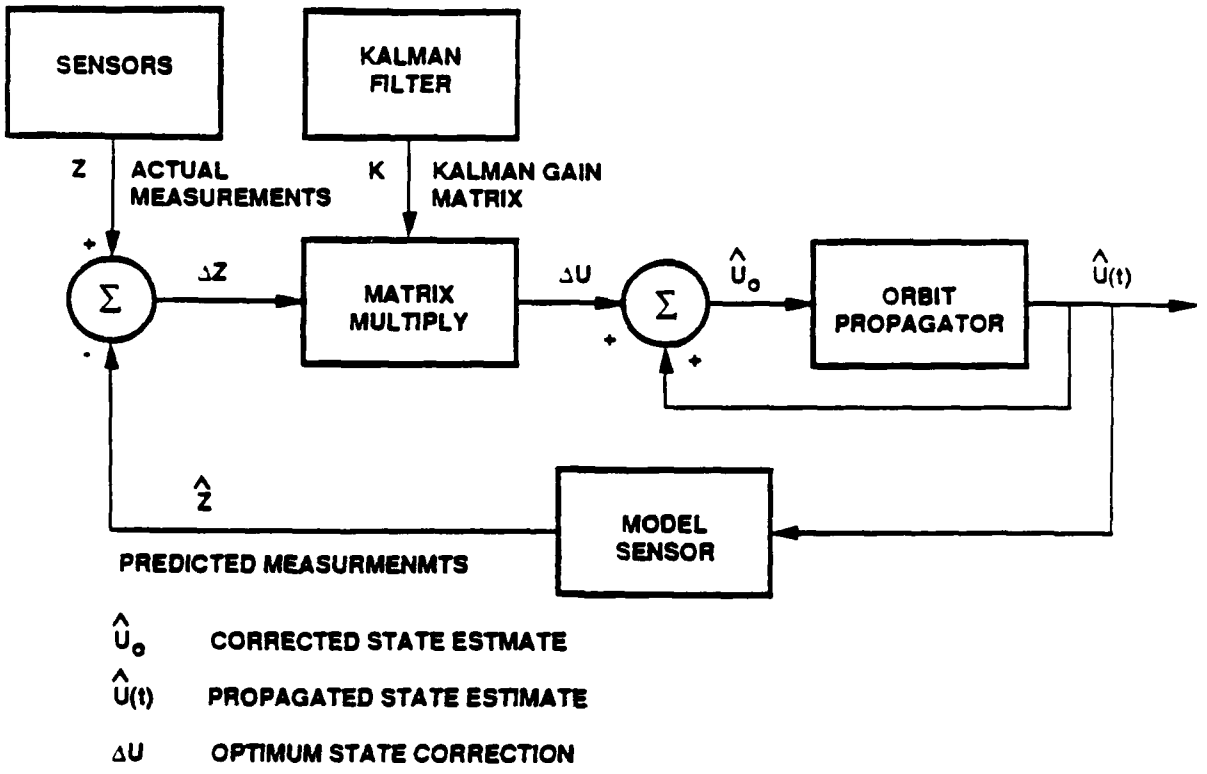

Figure 1: AutoNav System Block Diagram

2.2 Hardware

2.2.1 Sensor Description

The landmark sensor concept is based on the design of a strapdown star tracker similar to ones currently in use. The operation and performance requirements of the two sensors are essentially identical and, as discussed below, a similar set of design tradeoffs must be considered. A radiation-hardened version of such a star tracker has been built and tested as part of the parent development program for NRL. That breadboard employs a charge injection device (CID) detector, however, this sensor description also applies to designs using charge coupled device (CCD) detectors. The choice of a detector for this application would depend on the final system performance

requirements and environmental concerns, such as the radiation environment. The state of the art in detectors is dynamic.

The sensor concept, shown in Fig. 2, was developed as a general purpose standard star tracker (REF 1). A variation of this design was employed on the Retroreflector Field Tracker (RFT), which was used to track laser diode illuminated retroreflecting targets to measure the motion of a large flexible solar array (REF 2). This instrument flew on Space Shuttle mission 41-D in 1984.

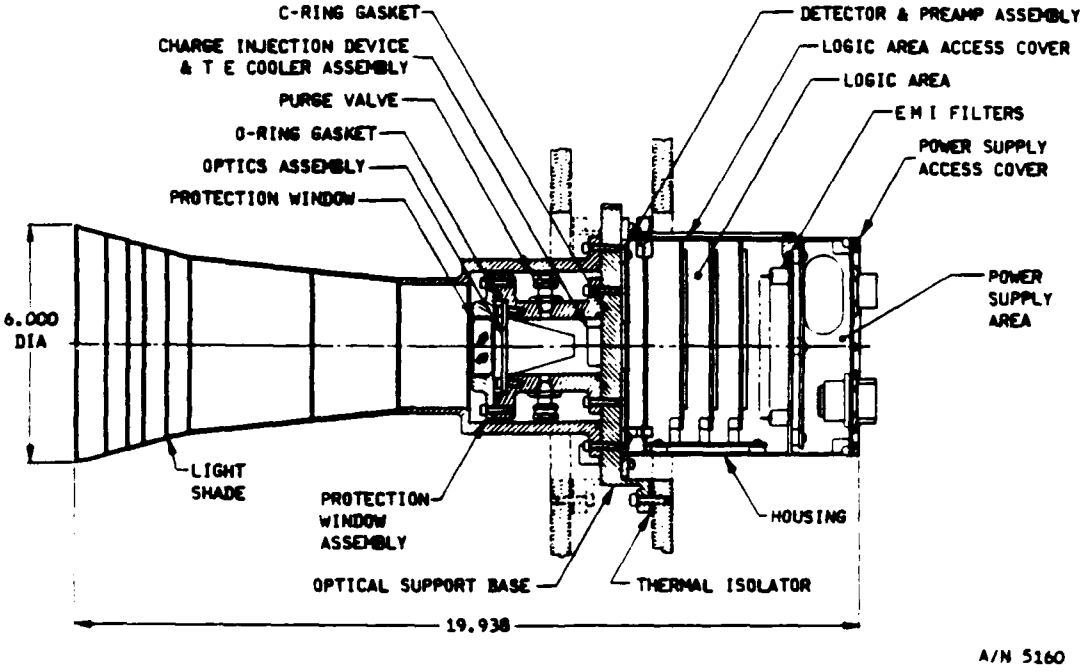

Figure 2: BASG Solid State Tracker

Typical performance and physical parameters for this concept are shown in Table 2. A range of parameters is shown to reflect that tradeoffs in field of view, integration time, detector format, target motion, optic size, accuracy, etc. exist and that the sensor configuration will reflect the final requirements. The range of values shown was used to develop the landmark sensor noise models used in the navigation filter simulation.

Field of View	$\leq 10\,\text{deg} \times 10\,\text{deg}$
Accuracy (temporal)	$1 - 10\,\widehat{\text{sec}}\ (1\sigma)$
Target Rate	$0 - 1\,\text{deg / sec}$
Target Brightness	$-8 - 0$ Mv (equivalent)
Volume	500 cu in
Weight	15 lb (6.8 kg)
Power	15 W

Table 1: Sensor Capabilities

The primary tradeoffs are between accuracy, field of view, and detector format. Constraints include target brightness, maximum target rate, and limitations on optics size.

The accuracy of the sensor defines the noise of the landmark measurement. Along with other sources, such as attitude determination uncertainty, it determines the noise input to the navigation filter. This study has considered total temporal noise inputs of 10, 20, and 30 $\widehat{\text{sec}}$ (1σ). In an actual system implementation this error would be apportioned to the various sensors and subsystems, but it is apparent that a landmark sensor accuracy on the order of that shown above is adequate.

Field of view (FOV) is a significant driver in the sensor design. Generally, a small field of view permits higher accuracy for a given detector format. However, a small field of view limits the area of the availability of targets, providing less frequent updates. Field of view then becomes a system trade that is made in the simulation of the navigation filter. This study has considered 10 and 20 degree fields of view. Given typical available CTD detector formats (i.e. 256 to 2048 pixels square) a 10 degree field of view sensor is reasonable for the accuracy range considered, while a 20 degree sensor would require use of the larger format detector and would provide an accuracy in the higher end of the range. The Retroreflector Field Tracker used a 256×256 pixel CID with a 22×22 degree field of view. It bettered its accuracy specification of 19 $\widehat{\text{sec}}$. Use of a larger format detector would improve accuracy.

The above discussion assumes a strapdown, or fixed, landmark sensor with its field of view centered on nadir. The effective field of view can be increased by using a one axis gimbal or a steering mirror to offset point the line of sight across the satellite ground track, making more targets available.

The current position estimate is used to predict upcoming targets and the sensor or line of sight is offset to provide the update. At low earth orbit the horizon may be up to 70 degrees from nadir and offset pointing would provide significantly more updates from a basic 10 degree sensor. The price for this is the increased complexity of the sensor and the addition of pointing errors. Again, the tradeoff between the frequency of updates and accuracy is made in the simulation of the navigation filter.

Target motion is an important driver for the sensor design. Accuracy improves with the time that the beacon signal is integrated on the detector. Orbital motion causes the target image to move on the detector, requiring the track logic to follow it and limiting the time that a pixel integrates the signal. Maximum motion occurs for a target at nadir, and varies from approximately 1 deg/sec for a 450 km orbit to 0.0 deg/sec for a geostationary orbit. These rates have been accommodated in star tracker designs, the typical tradeoff being in detector format: smaller formats produce a larger pixel angular field of view and allow a longer integration time, while larger formats have greater accuracy due to higher angular resolution.

One difference between a landmark tracker and a star tracker is that if a narrow band laser is used as a beacon source, a narrow spectral filter can be used to discriminate against the broad band stray light background. This technique was used on the Retroreflector Field Tracker to reduce reflected sunlight from the solar array.

2.2.2 Beacon Description

The landmark beacon configuration considered uses a near-infrared (approximately 800 nm wavelength) laser diode with an optical system to shape and project its beam. This configuration was demonstrated on the Retroreflector Field Tracker, in which two cylindrical lenses produced a fan-shaped beam which illuminated passive targets on the solar array. The design requirement is to provide a beam that will produce a starlight level brightness over an angular beamwidth matched to the sensor field of view.

Finite laser diode power levels provide a constraint on the brightness of the beacon source. Current high power laser diodes are available at the Watt level. Table 2 shows the brightness of a 1-Watt beacon in equivalent star magnitude for two beamwidths and several low earth orbit altitudes. The

Width	Altitude (km)	Magnitude
10 deg	400	+4.34
	635	+5.34
	1000	+6.33
20 deg	400	−5.84
	635	−6.84
	1000	−7.83

Table 2: Beacon Brightness (Meq)

beamwidths (total angle) match the two sensor fields of view used in the filter study and the 635 km altitude is the orbit simulated.

These levels are within the capability of the sensor. The use of higher powered diodes or an optical combination of several diodes or beacons would increase the received signal. Each time the power is doubled, the magnitude decreases (indicating a brighter source) by 0.75; for example, an 8 Watt beacon with a 10 degree beamwidth would appear to be a +3.1 M object. Large diode arrays are being developed as pumping sources for solid-state lasers, with powers up to 25 Watts having been reported. If this power were used, the above beacon would appear to be a +1.8 M object.

Clearly, a small beam angle increases beacon brightness and this is one factor that must be considered in sizing the sensor field of view. Other options may be considered to reduce beacon beamwidth; for instance, a fan-shaped beam. A beam sized to match the field of view in one dimension but narrower than the field of view in the other would provide the same target availability for orbital tracks crossing the beam perpendicularly. The target would appear, however, only during a portion of the pass, reducing the number of times the target could be sampled. Two such beams oriented orthogonal to each other would provide availability for any ground track. For instance, a 1-Watt, 2 deg × 10 deg beam would appear as a −3.9 M object, instead of the −5.34 M brightness shown above. An 8-Watt beacon would appear as −1.6 M, and a 25-Watt beacon, −0.4. Such an improvement in brightness and sensor accuracy would have to be traded against the reduced number of samples in the filter simulation.

	VAX 11/780	Sun 4/200	Amdahl 5860	MIPS
Dhrystone	1.0	10.9	16.4	23.9
LINPACK	1.0	7.9	—	25.7
Whetstone	1.0	4.0	—	16.4

Table 3: Relative Performance Comparisons of Processors

2.2.3 Computer Hardware

We did not, in this study, make an in-depth study of candidate flight computer systems. We merely recognized that the MIPS processor, which we are targeting for other flight systems, is more than equal to the task.

The MIPS is a 32 bit RISC machine which can be run at 25 MHz, but is more often operated at 10 MHz for better matching to currently available memory chips. It implements the IEEE 754 floating point standard, and performs F.P. additions and multiplications in 5 clock cycles, and divisions in 19 cycles. It incorporates a 5 step instruction pipeline, thereby attaining very impressive operation speeds. This is reflected in the following comparisons to other computer systems using several standard benchmarks.

The figures in Table 3 are for a MIPS running at 25 MHz, and indicate that the MIPS, on rough average, has 22 times the computing capabilities as the VAX. At 10 MHz, it *still* outperforms the VAX by a factor of 8.8.

A full compliment of program development tools is also available for the MIPS. The M120-3 development station comprises a 300 MB disk drive, 8 MB of RAM, and runs under the full UNIX operating system. Both an Assembler and a full optimizing C compiler are available, as well as the standard, indispensable, UNIX program development tools such as **grep**, **make**, **touch**, and **diff**.

The CPU uses 2.5 watts, and the Math Co-Processor about 5.0 watts. *Very* rough estimates of the power required by a system adequate for our needs puts the power at between 12 and 15 watts. If power must be limited, the Co-Processor could be eliminated, and the floating point computations done in software if the degradation in performance is acceptable.

For a fully contained MIPS processing system complete with boards, power supplies, and box, the weight will be about 10 pounds.

2.3 Algorithms & Software

2.3.1 Propagator

The propagator used derives from the assumption that the only force acting on the vehicle comes from the gravitational field of a homogeneous, oblate Earth. There is no air drag, solar pressure, nor magnetic effects. Thus, we have elliptical orbits which suffer drift in their ascending nodes and arguments of perigee. The "state", $\mathbf{u}$, of the vehicle is taken to be the standard Keplerian orbital elements. That is,

$$\mathbf{u} = \begin{bmatrix} a \\ e \\ i \\ \Omega \\ \omega \\ \nu \end{bmatrix} \tag{1}$$

These terms are identified in Appendix H.

The differential equations of motion for this choice of state vector are quite simple.

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}, t) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ d\Omega/dt \\ d\omega/dt \\ d\nu/dt \end{bmatrix} \tag{2}$$

where

$$\frac{d\Omega}{dt} = -\sqrt{\frac{\mu}{a^3}} \left(\frac{r_e}{a}\right)^2 \frac{3J_2}{2(1-e^2)} \cos i \tag{3}$$

$$\frac{d\omega}{dt} = -\sqrt{\frac{\mu}{a^3}} \left(\frac{r_e}{a}\right)^2 \frac{3J_2}{4(1-e^2)} \left[1 - 5\cos^2 i\right] \tag{4}$$

$$\frac{d\nu}{dt} = \frac{\sqrt{\mu}(1 + e\cos\nu)^2}{a(1-e^2)^{3/2}} \tag{5}$$

The propagator for this AutoNav system is merely a propagator for the true anomaly, $\nu(t)$. Given a value for t, the true anomaly is computed in several stages. First, compute the Mean Anomaly, M, as follows:

$$M = \frac{2\pi}{\tau}\left[(t - t_0)\mathrm{mod}(\tau)\right] \tag{6}$$

where t_0 is the time at epoch, and τ is the orbit period. Incidentally, the quantities t_0 and τ which appear in the above equation are

$$t_0 = t - \frac{\tau}{2\pi}\left[2\tan^{-1}\left(\sqrt{\frac{1-e}{1+e}}\tan\left(\frac{\nu}{2}\right)\right) - \frac{e\sqrt{1-e^2}\sin\nu}{1+e\cos\nu}\right] \tag{7}$$

and

$$\tau = 2\pi\sqrt{\frac{a^3}{\mu}} \tag{8}$$

Next, using this value for M, approximate the Eccentric Anomaly, E, by "solving" Kepler's nonlinear equation

$$M = E - e\sin E \tag{9}$$

Given E, the True Anomaly, ν, is computed using Gauss's equation.

$$\nu = 2\tan^{-1}\left\{\sqrt{\frac{1+e}{1-e}}\tan\left(\frac{E}{2}\right)\right\} \tag{10}$$

In the actual implementation of Eqs. 6, 9, and 10, care must be taken to keep computations within the appropriate principal value bounds. This has been done, but not detailed here in the interest of protecting the public welfare.

2.3.2 Kalman Filter

The basic equations of the Extended Kalman Filter are on display in Appendix B, and the way in which they are used is described there. Here, we will merely present the explicit form of certain of the terms specific to this problem.

Linearized Measurement Geometry, H

It is assumed that the output of the sensor(s) is preprocessed to provide the components of a *unit* vector, given in spacecraft coördinates, which points from the center of mass of the vehicle to the target. If the coördinates of the i-th reference point, in the Earth frame, E^e, are

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \tag{11}$$

then the vector, $\mathbf{v}_i$, from the spacecraft to this point is given by

$$\mathbf{v}_i = \mathbf{R}_1(-\frac{\pi}{2})\mathbf{R}_3(\nu + \frac{\pi}{2}) * \tag{12}$$

$$\left\{ \mathbf{R}_3(\omega)\mathbf{R}_1(i)\mathbf{R}_3(\Omega - \omega_e t - \theta_0) \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} - \frac{a(1-e^2)}{1 + e\cos\nu} \begin{bmatrix} \cos\nu \\ \sin\nu \\ 0 \end{bmatrix} \right\}$$

It is clear that this is not intended to be a unit vector. The associated *unit* vector, $\mathbf{g}_i$, is simply

$$\mathbf{g}_i = \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \tag{13}$$

and is what is produced by the sensor system.

The matrix $\mathbf{H}$ (Eq. 47 in Appendix B) is really the concatenation (stacking up) of the Jacobian of Eq. 13 with respect to $\mathbf{u}$ for as many reference points as are visible. That is,

$$\mathbf{H} = \begin{bmatrix} \frac{\partial \mathbf{g}_1}{\partial \mathbf{u}} \\ - - - \\ \vdots \\ - - - \\ \frac{\partial \mathbf{g}_k}{\partial \mathbf{u}} \end{bmatrix} \tag{14}$$

when there are k targets in view. In the early stages of the study, $\partial \mathbf{g}_k / \partial \mathbf{u}$ was computed numerically using double sided finite differences, but this was found to somewhat less stable than desired. It is also very computationally intensive. The Jacobian is, therefore, now computed analytically. This gives the additional benefit of reducing the execution time of the process. Actually,

$\partial \mathbf{g}/\partial \mathbf{u}$ itself is *not* computed. Obtaining these partials analytically would be *very* difficult, even with the aid of SMP[2]. Instead, $\partial \mathbf{v}/\partial \mathbf{u}$ is determined analytically (using SMP), and then transformed using Eq. 77, repeated here for convenience.

$$\mathbf{J}_g = \frac{\partial \mathbf{g}}{\partial \mathbf{u}} = \frac{1}{N}\left\{\mathbf{I} - \mathbf{g}\,\mathbf{g}^T\right\}\mathbf{J}_v \tag{15}$$

In this equation, $\mathbf{J}_v$ is the easily computed Jacobian of the non-unit-length measurement vector, $\mathbf{v}$; $\mathbf{g}$ is the unit vector parallel to $\mathbf{v}$; N is the magnitude of $\mathbf{v}$; $\mathbf{I}$ is the unit matrix; and $\mathbf{J}_v$ is the Jacobian of $\mathbf{g}$ ... which is required by the Kalman Filter. The analytical forms of $\mathbf{v}(\mathbf{u})$ and $\mathbf{J}_v$ are given in Appendix C.

Linearized Plant Geometry, F

The linearized plant geometry matrix, $\mathbf{F}$ (see Eq. 46, Appendix B) is defined as

$$\mathbf{F} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \tag{16}$$

where $\mathbf{f}(\mathbf{u}, t)$ is the nonlinear vector differential equation of motion, Eq. 2. This matrix is also formed analytically. All its components are zero except for the $\mathbf{F}_{i,j}$ listed below in which

$$\Upsilon \stackrel{def}{=} 3J_2\sqrt{\frac{\mu}{a^3}}\left(\frac{r_e}{a}\right)^2\frac{1}{(1-e^2)} \tag{17}$$

and

$$\frac{d\nu}{dt} = \frac{\sqrt{\mu}(1+e\cos\nu)^2}{a(1-e^2)^{3/2}} \tag{18}$$

are commonly occurring factors.

$$\mathbf{F}_{4,1} = -\frac{7\cos i}{4a}\Upsilon \tag{19}$$

$$\mathbf{F}_{4,2} = -\frac{e\cos i}{(1-e^2)}\Upsilon \tag{20}$$

[2]Symbolic Manipulation Program; an outgrowth of, and improvement upon, MIT's Macsyma program. SMP is the creation of Steve Wolfram.

$$\mathbf{F}_{4,3} = -\frac{\sin i}{2}\Upsilon \tag{21}$$

$$\mathbf{F}_{5,1} = -\frac{7(1-5\cos^2 i)}{8a}\Upsilon \tag{22}$$

$$\mathbf{F}_{5,2} = -\frac{e(1-5\cos^2 i)}{2(1-e^2)}\Upsilon \tag{23}$$

$$\mathbf{F}_{5,3} = -\frac{5\sin i\cos i}{2}\Upsilon \tag{24}$$

$$\mathbf{F}_{6,1} = -\frac{d\nu}{dt}\frac{3}{2a} \tag{25}$$

$$\mathbf{F}_{6,2} = -\frac{d\nu}{dt}\left(\frac{2\cos\nu}{1+e\cos\nu}+\frac{3e}{1-e^2}\right) \tag{26}$$

$$\mathbf{F}_{6,6} = -\frac{d\nu}{dt}\left(\frac{2e\sin\nu}{1+e\cos\nu}\right) \tag{27}$$

State Covariance Update, $\mathbf{P}_k(-)$

The state covariance update equation, Eq. 53, is used in this study. It is valid for *any* gain, $\mathbf{K}_k$, optimal or otherwise, and is reproduced here for reference.

$$\mathbf{P}_k(+) = \{\mathbf{I}-\mathbf{K}_k\mathbf{H}_k\}\,\mathbf{P}_k(-)\,\{\mathbf{I}-\mathbf{K}_k\mathbf{H}_k\}^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \tag{28}$$

If the *optimal* Kalman gain is used in Eq. 28, then that equation reduces to

$$\mathbf{P}_k(-) = \{\mathbf{I}-\mathbf{K}_k\mathbf{H}_k\}\,\mathbf{P}_k(-) \tag{29}$$

Although Eq. 29 is simpler and computationally faster than 28, it is prone to producing non-symmetric $\mathbf{P}_k(+)$ matrices. This is due solely to computational roundoff errors. Eq. 28, because of its form, is inherently symmetrical. It is felt that the freedom from worry about non-symmetric updates is worth the additional computation load of using Eq. 28. Reducing computation speed is not a high priority, right now, anyway. When this algorithm is committed to flight code, then it will be worth examining this matter to see if Eq. 29 can be safely used.

2.3.3 Computer Languages

The software for this entire project was written in the C language using the Microsoft C Compiler, Version 5.1. This language was chosen, primarily, for the following reasons:

1. It is ideally suited for scientific applications, and eclipses FORTRAN in these areas.

2. A vast store of support software is available within the C community which greatly aids and simplifies the software development process. No other language enjoys this support.

3. C was designed, at its inception, to be both an easily portable language, and to produce efficient executable code. Because of this, the code written for this project can be *very* easily moved onto a flight processor. BASG has experience in this area. We extracted the computational kernel of CLOP[3] for use as the real-time orbit propagator in one of our satellites[4]. The language is equally well suited for scientific and real-time applications.

2.3.4 Reference Points

The coördinates of the reference points (targets) used, in this study, for the Kalman update of the estimated state are contained in a list of 89 Air Force bases located in the political U.S.A. The east longitude, north latitude, and height above sea level for each base is specified. The file of these coördinates is given in Appendix G.

It is assumed that each of these bases houses a laser which can be seen by the sensor. See Sec. 2.2.2 for a fuller treatment of this subject.

REFERENCES:

[3]A general purpose, numerically integrating, orbit propagator developed at BASG, written by Charlie Ros5e, comprising high-order gravitational potentials, a variety of air-drag models, and designed to run on personal computers.

[4]The Remote Mirror Experiment, RME.

1. J. P. McQuerry & F. E. Wargocki, *"A User's Guide to the BASD Solid State Star Tracker"*, Guidance and Control 1985, R. D. Culp et. al., ed., Advances in the Astronautical Sciences, Vol. 57, p. 453 (1985)

2. F. E. Wargocki, A. J. Ray, and G. E. Hall, *"Retroreflector Field Tracker"*, State of the Art Imaging Arrays and Their Applications, K. Prettyjohns. ed., Proc. SPIE, p. 283 (1984)

3 AutoNav System: Evaluation

In this section, we describe how the AutoNav system was evaluated, and what simplifications were instituted in order to finish the study in finite time.

3.1 Overview

The AutoNav system, and the vehicle within which it lives, necessarily interact with the universe. The satellite is acted upon by the gravitational fields of the Earth, Moon, Sun, planets. It is buffeted about by solar pressure (both direct and reflected by the Earth), air drag, and magnetic fields for starters. The sensors produce noisy measurements, through a nonlinear transfer process, of reference points viewed through a shimmering, refractive medium. The final, and ultimate, evaluation of any AutoNav system's operation must take place in this environment.

Such real world tests, obviously and necessarily, are prohibitively expensive. They also do not permit experimentation with the algorithms. What is done, at this stage of the investigation, is to *simulate* the universe with as much fidelity as one can afford. The economics of the situation involve time, money, computing facilities, and the desired accuracies of the the results. A block diagram of the simulation used to evaluate the AutoNav system is shown in Fig. 3.

In a comprehensive evaluation, the following quantites would be simulated. Such a complete simulation was not warranted at this stage of our investigations, and the areas of departure are discussed in the following subsections.

- Forces on the vehicle. As mentioned earlier, they include those arising from the gravitational fields of many sources: air drag, magnetic forces if the vehicle is charged, and solar pressures both directed and reflected. The simulation of these forces is embodied in what we term the "Truth Propagator". This propagator must represent the actual motion of the vehicle as closely as possible, and is in contrast to the simpler "AutoNav Propagator" which forms part of the AutoNav system.

- The sensors. Measurement noises, nonlinearities, and mounting misalignments must be modeled.

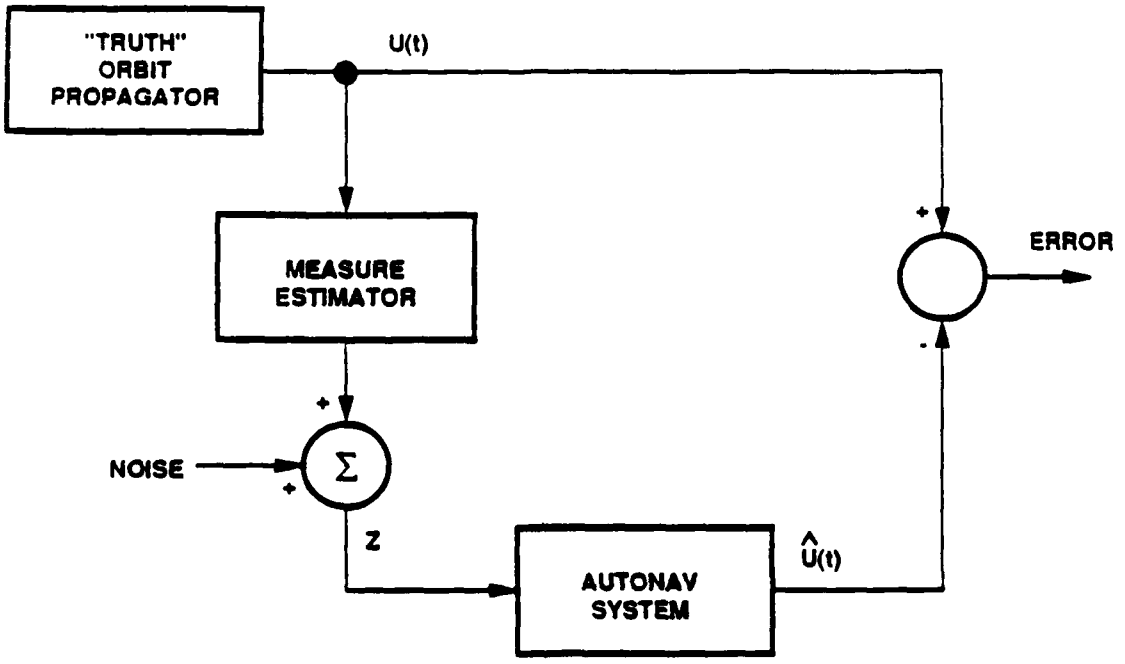

Figure 3: Simulation for AutoNav System Evaluation

- The reference points. Uncertainties in their surveyed positions may be modeled, as well as uncertainties in their observed positions due to atmospheric effects. These uncertainties comprise both random fluctuations, and slowly drifting biases.

- The vehicle attitude. This is important only for those AutoNav systems which make angular measurements upon the reference points. Ranging schemes are usually not plagued with such complications.

3.2 Landmarks

It was deemed appropriate, and realistic, to choose the Air Force bases located in the political United States as the locations of the reference points

Orbit Parameter	Value
a	7000.000 km
e	0.010
i	54°40′
Ω	359.984°
ω	0.011°
ν	56.413°

Table 4: Definition of Test Flight Orbit

which the star tracker would observe. Bases located in other parts of the world, especially Europe, would be the first to become unavailable at the precise time when the functioning of the AutoNav system became critical. The latitude, longitude, and height above sea-level of 89 such bases were placed in a data file, and accessed by the simulation program. A listing of this data file is included in Appendix G. The locations of the bases are shown in Fig. 4 on page 58.

Recall that the actual targets being observed are laser beacons located at these Air Force bases. To simplify the building of the simulator, it was assumed that each of these beacons radiated uniformly over a locally vertical cone having the same field of regard as the tracker. Thus, so long as the beacon's location was in the field of view of the tracker, the tracker responded. This is an over-simplification, and will be corrected in subsequent studies.

Uncertainties in the surveyed positions of the landmarks will be manifest as small angular offset biases on the order of 10 $\widehat{\text{sec}}$, and *must* be considered. However, they were ignored in this study because of the gross errors introduced by the AutoNav propagator. They will *not* be omitted in the future.

3.3 Orbit

An orbit with the following initial conditions, given as Keplerian elements, was chosen as being typical of the flights that could be expected.

The right ascension of the ascending node, Ω, was chosen to place the ground track of the orbit over the most A.F. bases along the east coast. This

gave us the most measurements possible; an important consideration given the difficulties we experienced with our choice of Model Propagator[5]. The ground track for this orbit is shown in Fig. 4. The locations of the A.F.bases are denoted by – signs.

3.4 Tracker Configuration

The AutoNav system included a single tracker having the characteristics described in Sec. 2.2.1. It was aligned to nadir to within the accuracy capabilities of an assumed attitude control system. This ACS was *not* modeled; its errors were assumed to be zero-mean with a Gaussian distribution, and were simply RSSed with the intrinsic beacon tracker noise. The same is true for other uncertainties in the measurement systems, such as "target twinkle" arising from atmospheric instabilities.

Constant, or very slowly varying, biases were ignored in this study. These included sensor mounting offsets, and the effects of atomospheric refraction. In these preliminary investigations, it was assumed that refraction can be predicted, and appropriate compensations made.

The field of view of the tracker could be specified as a run time parameter. Flights with different FOVs were made so it could be determined if a gimbaled mirror scanner would have to be added to the sensor system in order to see enough targets.

The tracker noise was also considered to be Gaussian with zero-mean. The noise, r_p, was simulated by using the following simple approximation to a Gaussian generator:

$$r_p = \mu_r + \sigma_r \sum_{k=1}^{12} (u_k - 0.5) \tag{30}$$

where σ_r and μ_r are the desired standard deviation and mean of the pseudo-random sequence, $\{r_p\}$, and u is a random variable which is uniformly distributed on the interval $[0,1]$. Granted: this is a kindergarten variety of Gaussian generator, but it was deemed adequate because our primary concerns lay in the gross behavior characteristics of the system.

[5]See Sec. F where this problem is discussed in detail.

3.5 Initial Covariance Matrices

The Kalman Filter incorporates three covariance matrices, the values of which affect the operation of the system. See Appendix B for a fuller description of these matrices.

3.5.1 State: P

This is a time varying matrix whose value is propagated by solving the differential equation, Eq. 50, and corrected occasionally by Eq. 53. Eq. 50 must be given an initial value for **P** at the beginning of the simulation, and, in lieu of good reasons for choosing otherwise, the following was used:

$$\mathbf{P}_0 = \begin{bmatrix} 2000 & 2000 & 2000 & 2000 & 2000 & 2000 \\ 2000 & 2000 & 2000 & 2000 & 2000 & 2000 \\ 2000 & 2000 & 2000 & 2000 & 2000 & 2000 \\ 2000 & 2000 & 2000 & 2000 & 2000 & 2000 \\ 2000 & 2000 & 2000 & 2000 & 2000 & 2000 \\ 2000 & 2000 & 2000 & 2000 & 2000 & 2000 \end{bmatrix} \tag{31}$$

3.5.2 Plant: Q

Often, the **Q** matrix is set to **0** for simplicity of operation, and because the analyst has no theoretical basis for choosing anything different. The result of this is that the gain matrix, **K**, goes to zero after a while, and the Kalman Filter becomes "conceited" by refusing to produce corrections to the estimate of the state.

We chose **Q** to be the following based on our desire to force the derivative of **P** to have a certain non-zero value when **P** was zero. This is what we used:

$$\mathbf{Q} = \begin{bmatrix} 10^{-7} & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^{-7} & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{-7} & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^{-9} & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^{-9} & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^{-9} \end{bmatrix} \tag{32}$$

3.5.3 Measurement: R

The dimension of **R** varies with the number of measurements being made. Therefore, it could not be specified as a particular thing at run time. We *did* define **R**, at each iteration, to be a constant diagonal matrix whose diagonal elements were the variance, σ^2, of the tracker noise.

3.6 Truth Propagator

The Truth Propagator was initially intended to be only the output of CLOP using a set of spherical harmonic zonal, sectoral, and tesseral coefficients of fairly high order. Although this *was* done, we found it necessary to use another, simpler propagator to verify the correct operation of the AutoNav system. This is described below.

Although CLOP has the capability of including air-drag forces, we flew all our simulations in vacuum. This is because the AutoNav propagator errors overshadowed all others, and air-drag effects would have been unnoticable.

3.6.1 High-Order Harmonic

The coefficients used to represent Absolute Universal Truth were obtained from

Journal of Geophysical Research
Volume 90, Number B11
September 30, 1985
Page 9288, Table 2
"LAGEOS Scientific Results"
Publisher: American Geophysical Union

Although coefficients through 36th order are given there, we used only those through order 16. We estimated that the effects of higher order terms would be miniscule for the flight durations we were considering.

3.6.2 Simple Truth Propagator

Attention! This subsection describes a simple test we made to verify the correct operation of the simulation. It served no other function than that!

Once we realized that an actual orbit differed *substantially* from one represented by Keplerian elements in which Ω and ω drifted at constant rates given by Eq. 3 and 4, we temporarily replaced the accurate, CLOP propagator with the Keplerian AutoNav propagator. This permitted us to evaluate system performance in the absence of divergence between propagators, to validate the Kalman Filter, and to gain basic insight into the operation of the system as a whole.

The filter brought the initial error from a kilometre down to a millimetre very quickly, thereby giving us confidence in the correctness of the system as a whole.

3.7 System Performance

We evaluated the AutoNav system performance by making a series of simulated flights over the eastern United States. This ensured that the sensor would see enough targets to keep the Truth and AutoNav propagators from diverging too much. Plots of the position errors for the various flights appear figures 6 through 17, and the conditions of the flights are summarized in the following table. The reader is cautioned that the vertical axis scales on these plots are almost all different. We accepted the automatic scaling of the graphing package to show the maximum detail in the plots.

In the Truth Model column of the following table, the term "Keplerian" means that the Approximate Truth propagator described in Sec. 3.6.2 was used. The term "16th" denotes the 16th order spherical harmonic model based on the LAGEOS gravitational potential model.

The position error plots for the various flights are probably self-explanatory, but a few observations can be made. The reader will likely notice that, in many cases, even though the position error has been reduced to very low levels, the position error begins to increase substantially when measurements are no longer available. This is because the *velocity* error results in an estimated orbit which has a slightly different inclination than the real orbit.

Flight	FOV	1σ Noise	Truth Model	Init. Error
1	10°	0 sec	Keplerian	100 m
1a	20°	0 sec	Keplerian	100 m
1b	20°	0 sec	Keplerian	100 m
1c	20°	0 sec	Keplerian	2000 m
2	10°	0 sec	16th	0 m
2a	20°	0 sec	16th	0 m
3	10°	10 sec	16th	0 m
3a	20°	10 sec	16th	0 m
3b	10°	20 sec	16th	0 m
3c	20°	20 sec	16th	0 m
3d	10°	30 sec	16th	0 m
3e	20°	30 sec	16th	0 m

Table 5: Synopsis of Flight Parameters

Hence, the estimated position begins to draw away from the real position.

Small $\hat{\mathbf{u}}$ Error Fig. 5 shows how the distance between two vehicles changes when their initial positions and velocities are slightly different. For the case shown, the position error was

$$\Delta\mathbf{p} = \begin{bmatrix} 1.3 \\ 1.9 \\ 1.9 \end{bmatrix} \; metres \tag{33}$$

and

$$\Delta\mathbf{v} = \begin{bmatrix} 0.0 \\ 0.1 \\ 0.1 \end{bmatrix} \; metres \quad sec^{-1} \tag{34}$$

Note the axes' units. The vertical axis is in kilometres, while the horizontal axis is in thousands of seconds. Thus, after some three orbits, this very small initial error grown to about 6 kilometres.

Kepler Representation Error The plot shown in Fig. 18 on page 65 shows how the position of a vehicle, as predicted by the Keplerian formulation with drifting Ω and ω, departs from an actual orbit with which

it initially matches. In this case, the "actual orbit" is derived by using a sixteenth order geopotential model to compute the gravitational forces, an all other forces are ignored. For a near-circular orbit having a 54.666° inclination, and 100 min period, the Keplerian prediction is in error by some 16 km after a single orbit. This indicates that it is a poor candidate for an AutoNav propagator.

Flight 1 The AutoNav system was given a 100 m initial error. Forty three measurements on several targets were made during the flight, and it is seen that the error was continually reduced until no more targets could be seen. Then, since the Truth and AutoNav orbits were slightly displaced despite their identical analytical *form*, the true and estimated positions began to diverge. (Fig. 6)

Flight 1a This is the same as Flight 1, except that the sensor's FOV is 20 degrees. Now 264 measurements were made on about ten different beacons, and the initial error was reduced substantially. The very slight residual error remaining after about 275 seconds was still enough to cause ultimate divergence. (Fig. 7)

Flight 1b This is the same as Flight 1a, but shows behavior over a full orbit, rather than just the time during which measurements are available. (Fig. 8)

Flight 1c Again, essentially the same as Flight 1, but with an initial position error of 2 kilometres. The purpose of this flight was to show that the Kalman Filter could, indeed, correct the AutoNav propagator after the system had flown with measurements for a full orbit. (Fig. 9)

Flight 2 Here, the 16th order LAGEOS gravitational potential was used for the truth model. The large difference between it and the AutoNav Keplerian propagator is evident in the very large error growth rates; roughly 3.5 metres/sec when no measurements are available. (Fig. 10)

Flight 2a The same as Flight 2, but with a 20 degree sensor FOV. Again: large error growth rates are seen in the absence of measurements but, because of the larger aspect angle, 265 measurements were made which kept the maximum error lower. (Fig. 11)

The remaining flights were made with the 16th order LAGEOS model, and the matrix of FOVs and sensor noises shown in the table. They were *intended* to give us insight into the sensor and attitude control system uncertainties that could be tolerated for acceptable navigation performance. Unfortunately, the inadequacy of the AutoNav propagator caused estimation errors which overshadowed the effects of the sensor noise.

4 Conclusions

The question of whether a beacon tracker, similar to the NRL Rad-Hard Star Tracker, can be successfully used in a Landmark Tracking AutoNav application rests squarely on the accuracies required, the number of beacons available, and their optical characteristics as well as distribution over the surface of the Earth. Basically, we have found that a star tracker *can* be used in a landmark approach, but only within some rather severe constraints:

- Beacons can be distributed more–or–less uniformly over the Earth. Restriction to the United States does not seem feasible.
- Position errors of several kilometres can be tolerated between updates.
- A very sophisticated orbit propagator is built into the system.

There are non-trivial problems associated with this approach to the problem. Most of them stem from the fact that the tracker would be used in a way for which it was not designed nor intended.

- The beacons "twinkle".
- The light from the beacons is refracted by the atmosphere by varying amounts which can be only partially compensated for.
- The tracker's field of view is too small: it greatly reduces the number of targets that can be seen, as well as the durations of observations. This results in fewer updates, and necessitates a *very* sophisticated orbit propagator. If the FOV is increased optically, the measurement errors increase in proportion. If it is expanded by a gimbaled "steering mirror", the mechanical complexity is increased greatly, as is the system weight. Accuracies also suffer. When the FOV is enlarged by *any* means, there is an attendant increase in the complexity of the beacons themselves. Their field of radiation must be enlarged accordingly.
- The already too few beacons can be obscured by cloud cover. This is, perhaps, the most egregious problem with this approach.

4.1 Why The Keplerian AutoNav Propagator?

The question naturally arises as to why we chose to use the Keplerian formulation, with Ω and ω drifting at the rates predicted by the First Order Theory, for the AutoNav propagator. It was done for two reasons.

Simplicity A guiding principle in designs of this kind is to make the problem as simple as possible. By choosing the Keplerian representation, rather than the Cartesian, we hoped to reduce the computational load on the flight computer, speed up the operation, and reduce the size and complexity of the hardware.

Ignorance We did not realize, in the beginning, that the actual orbital motion would depart so drastically from the First Order Theory as it did. We knew that there would be *some* difference between reality and the model, but did not suspect that it would be this great. If we had, we would have gone to the Cartesian representation (with its ability to easily handle complex forces from various sources) at the outset, and accepted the cost of computational complexity.

4.2 Suggestions for Further Study

With some enhancement of already existing simulation software, we will be able to answer, with confidence, the questions of accuracies and hardware requirements of an optical, landmark tracking AutoNav system. Such enhancements would include the following:

- Replace the simplistic Keplerian propagator in the AutoNav system with one which solves the differential equations of motion numerically. This involves only lifting the computational kernel from CLOP, which already exists. This propagator would be formulated in terms of Cartesian coördinates, not the Keplerian orbital elements. (See Sec. F)

- Casting the AutoNav propagator in terms of Cartesian coördinates will necessitate reformulating the linearized plant and measurement matrices, $\mathbf{F}$ and $\mathbf{H}$, for the Kalman Filter. This is not difficult, and will result in simpler computations. We also suspect that somewhat lower state estimation errors will result from such a formulation, but it is only that: a suspicion.

- As a means of refining the Truth Model, add a sophisticated model of air density to CLOP. This will probably be one of the Jacchia-Roberts models, with which we already have some familiarity. These are dynamic models which permit the density to vary with the relative position of the Sun.

- Add the position of the sun to the simulator truth model. This will permit day–night discriminations to be made when deciding if a given target is visible. The software module to do this already exists.

- Include bias errors such as sensor mounting position, sensor signal offsets, reference point location, and atmospheric refraction in the parametric studies.

- Perform a comprehensive sensitivity analysis of the system which will tell us the effects of: simplifying the AutoNav gravitational potential model, air drag, bias errors in various parts of the system, etc.

- Recast the entire AutoNav system in terms of using *range*, rather than *angular*, measurements to beacons located either on the ground or on coöperative satellites in (probably) synchronous Earth orbit. We feel, for a variety of reasons, that such an approach would provide a simpler, more accurate, and more reliable solution to the DOD AutoNav problem.

This Page Left Blank On Purpose

Appendices

A Notation

The symbol E is used to represent an orthogonal, right-handed, coördinate system in 3-space. Superscripts distinguish one system from another, while subscripts denote coördinate axes. Thus, E^q specifies the q coördinate system, while E_2^q denotes the 2-axis of coördinate system q. The subscripts can only have values 1, 2, and 3.

Vectors are denoted by boldface lower case letters, while matrices are boldface upper case. Thus, the product of vector $\mathbf{v}$ by matrix $\mathbf{M}$ to produce vector $\mathbf{z}$ is written as

$$\mathbf{z} = \mathbf{M}\mathbf{v} \tag{35}$$

The term $\mathbf{R}_k(\theta)$ represents an orthogonal transformation (rotation) of a coördinate frame about its k axis through an angle θ. Such rotations are right-handed. Thus,

$$\mathbf{R}_1(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{36}$$

$$\mathbf{R}_2(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{37}$$

$$\mathbf{R}_3(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{38}$$

So, if coördinate system E^b is produced by rotating coördinate system E^a about E^a's 2-axis through an angle β, this transformation is expressed as

$$E^b = \mathbf{R}_2(\beta)E^a \tag{39}$$

Next, suppose we have a vector, $\mathbf{r}$, whose components in E^a are given as $\mathbf{r}^a$. Its components in E^b may then be computed as

$$\mathbf{r}^b = \mathbf{R}_2(\beta)\mathbf{r}^a \tag{40}$$

A sequence of rotations is expressed as an *ordered* sequence of products of these rotation matrices. For example, a rotation about the 1-axis through

an angle θ_1 followed by a rotation about the new 3-axis through and angle θ_2 is written as

$$\mathbf{T} = \mathbf{R}_3(\theta_2)\mathbf{R}_1(\theta_1) \tag{41}$$

Remember that the order of these terms is crucial!

An obvious identity obtains from the fact that two sequential rotations about the *same* axis is the same as a single rotation about that axis through the sum of the individual angles. Thus,

$$\mathbf{R}_k(\alpha)\mathbf{R}_k(\beta) \equiv \mathbf{R}_k(\alpha - \beta) \tag{42}$$

Since rotation matrices have the special property that

$$(\mathbf{R}_k(\alpha))^{-1} \equiv \mathbf{R}_k(-\alpha) \tag{43}$$

the inverse of a sequence of rotations can be expressed simply by reversing the order of mulitplication, and changing the signs of their angles. That is,

$$(\mathbf{R}_j(\alpha)\mathbf{R}_k(\beta))^{-1} \equiv \mathbf{R}_k(-\beta)\mathbf{R}_j(-\alpha) \tag{44}$$

and so forth.

B Extended Kalman Filters

This section is merely a presentation, and a minimal elucidation, of the equations of the Extended Kalman Filter. No attempt to *derive* the equations is made. If the reader is interested in detail at that level, there are a multitude of references where s/he may look.[7]

The following terms appear in the Kalman equations:

Symbol	Description	Dimension
n	Number of state vector components	
m	Number of measurements	
$\mathbf{u}$	State vector	n
$\mathbf{z}$	Measurements vector	m
$\mathbf{P}$	State covariance	$n \times n$
$\mathbf{R}$	Measurements covariance	$m \times m$
$\mathbf{Q}$	Plant covariance	$n \times n$
$\mathbf{K}$	Kalman gain	$n \times m$
$\mathbf{F}$	Linearized plant geometry	$n \times n$
$\mathbf{H}$	Linearized measurement geometry	$m \times n$

The following compact notation is also used:

$$q(\pm) \stackrel{def}{=} q(t_k \pm \delta t) \tag{45}$$

where δt is an infinitesimal time interval, and the subscript k denotes a variable's value at the k-th sample time.

Let $\mathbf{h}(\mathbf{u}, t)$ be the (probably) nonlinear measurement prediction, and let $\mathbf{f}(\mathbf{u}, t)$ be the (almost assuredly) nonlinear differential equations of motion. Furthermore, define the Jacobians of $\mathbf{f}$ and $\mathbf{h}$ to be

$$\mathbf{F} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \tag{46}$$

$$\mathbf{H} = \frac{\partial \mathbf{h}}{\partial \mathbf{u}} \tag{47}$$

[7] See, for instance, "Applied Optimal Optimization", Arthur Gelb, Editor, MIT Press, 1974. Chapter 6, page 180, treats nonlinear estimation.

That is,

$$[\mathbf{F}]_{jk} = \frac{\partial f_j}{\partial u_k} \tag{48}$$

and similarly for $\mathbf{H}$.

During periods when no measurements are available, the state vector and the state covariance matrix are propagated forward in time by solving the following differential equations:

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}, t) \tag{49}$$

$$\frac{d\mathbf{P}}{dt} = \mathbf{FP} + \mathbf{PF}^T + \mathbf{Q}(\mathbf{u}, t) \tag{50}$$

These two differential equations are solved[8] in the interval

$$t_{k-1} \leq t < t_k \tag{51}$$

At time t_k, when a measurement set *is* available, those measurements are processed using the following Kalman equations:

$$\mathbf{K}_k = \mathbf{P}_k(-)\mathbf{H}_k^T \left\{\mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k^T + \mathbf{R}_k\right\}^{-1} \tag{52}$$

$$\mathbf{P}_k(+) = \{\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\}\,\mathbf{P}_k(-)\,\{\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\}^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \tag{53}$$

$$\mathbf{u}_k(+) = \mathbf{u}_k(-) + \mathbf{K}_k\{\mathbf{z}_k - \mathbf{h}(\mathbf{u}_k(-), t)\} \tag{54}$$

Equations 53 and 54 are then used as initial conditions for 49 and 50 over the next dry spell of no measurements.

[8]Solved, in all likelihood, by using numerical algorithms such as Runge-Kutta or Bulirsch-Stör.

C Analytical Expressions for v and $\partial \mathbf{v}/\partial \mathbf{u}$

This appendix presents the analytical expressions for $\mathbf{v}(\mathbf{u})$ and $\partial \mathbf{v}/\partial \mathbf{u}$. They have been lifted, bodily, from the SMP log files in which they were derived. They are, therefore, functionally correct, but not pretty.

The following are a set of replacement rules (variable name substitutions) that are pertinent to the rest of the appendix. It should be noted that

`Omega`	$\longrightarrow$	Ω
`omega`	$\longrightarrow$	ω
`we`	$\longrightarrow$	ω_e
`t`	$\longrightarrow$	t
`i`	$\longrightarrow$	i
`nu`	$\longrightarrow$	ν
`Cos[]`	$\longrightarrow$	cos()
`Sin[]`	$\longrightarrow$	sin()
`v[k]`	$\longrightarrow$	$\mathbf{v}_k$
`jv[j,k]`	$\longrightarrow$	$[\mathbf{J}_v]_{jk}$

```
(Cos[Omega - t*we - theta0] -> c1)
(Sin[Omega - t*we - theta0] -> s1)
(Cos[nu + omega]            -> c2)
(Sin[nu + omega]            -> s2)
(Cos[i]                     -> c3)
(Sin[i]                     -> s3)
(Cos[nu]                    -> c4)
(Sin[nu]                    -> s4)
(Cos[omega]                 -> c5)
(Sin[omega]                 -> s5)

(s1*s2 - c1*c2*c3           -> term1)
(c1*s2 + c2*c3*s1           -> term2)
(c1*c2 - c3*s1*s2           -> term3)
(c2*s1 + c1*c3*s2           -> term4)
```

The vector, $\mathbf{v}$, from the vehicle to the reference point at $[x, y, z]^T$ is given by

```
v[1] : -(term1*y + term2*x - (c2*s3*z))

v[2] : -(c3*z - (s3*(c1*y - (s1*x))))

v[3] : (a*(1 - e^2))/(1 + c4*e) - (term3*x) - (term4*y) -
       (s2*s3*z)
```

As noted earlier, this is *not* a unit vector.

The components of the Jacobian, $\partial \mathbf{v}/\partial \mathbf{u}$, are

```
jv[1,1] : 0

jv[1,2] : 0

jv[1,3] : c2*(c3*z + s3*(-(c1*y) + s1*x))

jv[1,4] : term1*x - (term2*y)

jv[1,5] : -(term3*x + term4*y + s2*s3*z)

jv[1,6] : -(term3*x + term4*y + s3*z*(c4*s5 + c5*s4))

jv[2,1] : 0

jv[2,2] : 0

jv[2,3] : c3*(c1*y - (s1*x)) + s3*z

jv[2,4] : s3*(-(c1*x) - (s1*y))

jv[2,5] : 0
```

```
jv[2,6] : 0

jv[3,1] : (1 - e^2)/(1 + c4*e)

jv[3,2] : -((a*(2e + c4*(1 + e^2)))/(1 + c4*e)^2)

jv[3,3] : -(s2*(c3*z + s3*(-(c1*y) + s1*x)))

jv[3,4] : -(term3*y) + term4*x

jv[3,5] : term1*y + term2*x - (c2*s3*z)

jv[3,6] : -((a*e*s4*(-1 + e^2))/(1 + c4*e)^2) + term1*y +
           term2*x - (c2*s3*z)
```

D Transformation of Jacobians

Here is the problem. Suppose you have the m-vector $\mathbf{v}$ which is a function of the n-vector $\mathbf{x}$, and further suppose that you find it necessary to normalize $\mathbf{v}(\mathbf{x})$ to unit length, and find the Jacobian of the result. That is, you must compute $\partial \mathbf{g}/\partial \mathbf{x}$ where

$$\mathbf{g}(\mathbf{x}) \stackrel{def}{=} \frac{\mathbf{v}(\mathbf{x})}{|\mathbf{v}|} \tag{55}$$

$$= \frac{\mathbf{v}(\mathbf{x})}{N(\mathbf{x})} \tag{56}$$

Note that $\mathbf{g}$ is forced to be a unit vector, the magnitude of which is independent of $\mathbf{x}$. We desire the Jacobian of $\mathbf{g}$, but it will probably be virtually imposible to compute analytically. We *are* willing and able, however, to compute the Jacobian of $\mathbf{v}$. The question is:

> "Is it possible to obtain $\partial \mathbf{g}/\partial \mathbf{x}$ through a reasonable transformation of $\partial \mathbf{v}/\partial \mathbf{x}$?"

We can, indeed, do this, and this section shows how.

Define

$$\mathbf{J}_v = \frac{\partial \mathbf{v}}{\partial \mathbf{x}} \tag{57}$$

and

$$\mathbf{J}_g = \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \tag{58}$$

Now, differentiating Eq. 56, we get

$$\frac{\partial g_i}{\partial x_j} = \frac{1}{N^2}\left(N\frac{\partial v_i}{\partial x_j} - v_i\frac{\partial N}{\partial x_j}\right) \tag{59}$$

$$= \frac{1}{N}\frac{\partial v_i}{\partial x_j} - \frac{1}{N}\frac{v_i}{N}\frac{\partial N}{\partial x_j} \tag{60}$$

$$= \frac{1}{N}\left\{\frac{\partial v_i}{\partial x_j} - g_i \frac{\partial N}{\partial x_j}\right\} \tag{61}$$

Now remember; N is the Euclidean norm of $\mathbf{v}(\mathbf{x})$. That is,

$$N = \left[\sum_{k=1}^{m} v_k^2\right]^{1/2} \tag{62}$$

so

$$\frac{\partial N}{\partial x_j} = \frac{1}{2}\left[\sum_{k=1}^{m} v_k^2\right]^{-1/2} 2\sum_{k=1}^{m} v_k \frac{\partial v_k}{\partial x_j} \tag{63}$$

$$= \frac{1}{N}\sum_{k=1}^{m} v_k \frac{\partial v_k}{\partial x_j} \tag{64}$$

$$= \sum_{k=1}^{m} g_k \frac{\partial v_k}{\partial x_j} \tag{65}$$

Combining this with Eq. 61 gives

$$\frac{\partial g_i}{\partial x_j} = \frac{1}{N}\left\{\frac{\partial v_i}{\partial x_j} - g_i\left(\sum_{k=1}^{m} \frac{\partial v_k}{\partial x_j} g_k\right)\right\} \tag{66}$$

This is a nice, and moderately useful, result but it admits of much further simplification.

Remember: from Eqs. 57 and 58,

$$\mathbf{J}_{v_{ij}} = \frac{\partial v_i}{\partial x_j} \tag{67}$$

and

$$\mathbf{J}_{g_{ij}} = \frac{\partial g_i}{\partial x_j} \tag{68}$$

We can pluck the summation term out of Eq. 66, and write it as

$$\sum_{k=1}^{m} \frac{\partial v_k}{\partial x_j} g_k = \sum_{k=1}^{m} \mathbf{J}_{v_{kj}} g_k \tag{69}$$

$$= \sum_{k=1}^{m} \left[\mathbf{J}_v^T\right]_{jk} g_k \tag{70}$$

$$= \left[\mathbf{J}_v^T \mathbf{g}\right]_j \tag{71}$$

Now, the term $\mathbf{J}_v^T\mathbf{g}$ is an n-vector, of which Eq. 71 is the j-th component. For the nonce, let

$$\mathbf{p} \stackrel{def}{=} \mathbf{J}_v^T \mathbf{g} \tag{72}$$

Then Eq. 66 becomes

$$\frac{\partial g_i}{\partial x_j} = \frac{1}{N}\left\{\frac{\partial v_i}{\partial x_j} - g_i p_j\right\} \tag{73}$$

or, in terms of matrices instead of their components,

$$\mathbf{J}_g = \frac{1}{N}\left\{\mathbf{J}_v - \mathbf{g}\mathbf{p}^T\right\} \tag{74}$$

$$= \frac{1}{N}\left\{\mathbf{J}_v - \mathbf{g}\left(\mathbf{J}_v^T\mathbf{g}\right)^T\right\} \tag{75}$$

$$= \frac{1}{N}\left\{\mathbf{J}_v - \mathbf{g}\,\mathbf{g}^T\mathbf{J}_v\right\} \tag{76}$$

$$\mathbf{J}_g = \frac{1}{N}\left\{\mathbf{I} - \mathbf{g}\,\mathbf{g}^T\right\}\mathbf{J}_v \tag{77}$$

Eq. 77 is the result sought. Is it not truly fine?

E The Truth Model

This appendix was extracted, virtually verbatim, from the CLOP User's Manual, and appears here merely as reference material for the reader who may not be familiar with how propagators of this variety work.

Since the following two subsections were not written expressly for this report, this is a little, minor, inconsistency in the naming of variables. For example, in Sec. E.1, the symbol **f** is used to mean "force on the vehicle", whereas in the rest of the report, it means "nonlinear differential equations of motion". The reader is beseeched to graciously overlook these shortcuts that were taken in the interest of expediency.

E.1 Differential Equations of Motion

The entire CLOP program is based on the solution of the ubiquitous vector differential equation

$$\mathbf{f} = m\mathbf{a} \tag{78}$$

which can be rewritten as

$$\mathbf{a} = \ddot{\mathbf{r}} = \frac{\mathbf{f}}{m} \tag{79}$$

where, of course,

$$\ddot{\mathbf{r}} = \frac{d^2\mathbf{r}}{dt^2} \tag{80}$$

In component form,

$$\ddot{\mathbf{r}} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \tag{81}$$

If the forces, f_k, in this equation are known, then it may be integrated to produce the position and velocity of the vehicle at any time. At least, this is possible in principle. It is even possible, in practice, if the forces are simple enough. For instance, if the force **f** results from the gravitational field of a spherically symmetrical mass distribution, then it has the form

$$\mathbf{f} = -\frac{1}{r^3} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{82}$$

and Eq. 81 can be integrated to produce the tidy elliptical orbits of freshman physics classes. When **f** becomes more complicated than this, as a result of modeling non-symmetric mass distributions and horrific atmospheric drags, things become intractable *fast*. Then, Eq. 81 can no longer be solved analytically. Numerical methods must be used to obtain approximate solutions. This is what is done in CLOP.

The primary algorithm used by CLOP to solve differential equations numerically is an 8th order Runge-Kutta process although a 4th order is also available. It is naturally suited to solving systems of 1st order simultaneous differential equations; not the 2nd order set shown in Eq. 81. To accommodate R-K, we define a six component System State Vector, **u**, as follows:

$$\mathbf{u} \stackrel{def}{=} \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \mathbf{r} \\ \dot{\mathbf{r}} \end{bmatrix} \tag{83}$$

The elements of **u** are the components of the vehicle's position and velocity expressed in E^c. Using this vector, Eq. 81 can be rewritten as

$$\frac{d\mathbf{u}}{dt} = \mathbf{A}\mathbf{u} - \mathbf{B}(\mathbf{a}_g - \mathbf{a}_d) \tag{84}$$

where $\mathbf{a}_g$ and $\mathbf{a}_d$ are the accelerations of the vehicle resulting from the Earth's gravitational field, and atmospheric drag, respectively. In Eq. 84, **A** and **B** are constant matrices. The expanded form of Eq. 84, showing **A** and **B**, is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} a_{gx} \\ a_{gy} \\ a_{gz} \end{bmatrix} - \begin{bmatrix} a_{dx} \\ a_{dy} \\ a_{dz} \end{bmatrix} \right) \tag{85}$$

The acceleration forcing terms appearing in equation 84 are discussed in the following sections.

It should be noted that Eq. 84 does not contain all the multitudinous terms which can affect the acceleration of the vehicle. As was mentioned in this manual's introduction, the effects of solar wind, Earth albedo, magnetic forces on a charged satellite, and the gravitational forces of the Sun, Moon, and planets have not been included. Relativistic and astrologic effects have also been ignored.

E.2 Gravitational Model

The acceleration a body experiences, as a result of the gravitational field in which it is immersed, is a vector quantity. Therefore, any mathematical model of the gravitational field must be a vector model. But the gravitational field of the Earth (and virtually any other body about which one might be interested in studying satellite orbits) is sublimely complex, and simple models of the field will just not do. The point is this; the mathematical representation of a complex vector field gets out of hand quickly.

Mathematicians, being a basically lazy lot, have come up with a simplification. It is well known that the gradient of a *scalar* function of position produces a vector.[9] So, if we can construct a scalar function, say $U(r, \phi, \lambda)$, of position such that its gradient matches the gravitational field of interest, then we will have only one messy equation to tote about, and life will be much easier. Granted, the gradient of this function will have to be computed if ever the function is to be actually *used* for anything, but that's a mere detail not pertinent to the theory of potential functions.

This scalar function of position is commonly called a *Potential Function*. I suspect the reason for this stems from the fact that the potential energy imparted to a particle by moving it through a force field, **f**, is

$$U = \int_S \mathbf{f} \bullet d\mathbf{s} \tag{86}$$

and that the gradient of this potential energy extracts the force; a vector quantity.

We follow the same convention here, except that, as the implementors, we must compute the gradient. This is actually done analytically within CLOP;

[9] This is a matter of definition, actually.

numerical differentiation is *not* done. The gravitational potential, U, and its gradient are treated in the following sections.

E.2.1 The Gravitational Potential, $U(r, \phi, \lambda)$

The gravitational acceleration term, $\mathbf{a}_g$, is derived as the gradient of a scalar potential function, U. That is,

$$\mathbf{a}_g = \nabla U \tag{87}$$

In rectangular coördinates, the gradient is

$$\nabla U^c = \begin{bmatrix} \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \\ \frac{\partial U}{\partial z} \end{bmatrix} \tag{88}$$

Because of the spherical symmetry of the situation, however, the potential function is most easily expressed in terms of local spherical coördinates, in which case the gradient becomes

$$\nabla U^l = \begin{bmatrix} \frac{\partial U}{\partial r} \\ \frac{1}{r\cos(\phi)} \frac{\partial U}{\partial \lambda} \\ \frac{1}{r} \frac{\partial U}{\partial \phi} \end{bmatrix} \tag{89}$$

In this form, r is the radial distance from the center of the gravitating body, λ is the *east* longitude, and ϕ is the *north geocentric* latitude.

The potential function, expressed in spherical coördinates *at the point where the vehicle is located*, is

$$U = \frac{\mu}{r}\left[1 - \sum_{n=2}^{N}\left(\frac{r_e}{r}\right)^n \sum_{m=0}^{n} P_n^m(s\phi)\{C_{nm}\cos(m\lambda) + S_{nm}\sin(m\lambda)\}\right] \tag{90}$$

where

$$s\phi \stackrel{def}{=} \sin(\phi) \tag{91}$$

The $P_n^m(s\phi)$ terms are the *Legendre Associated Functions* of degree n and order m. These functions are treated more fully in Section E.2.3.

Equation 90 is not the result of a late night fancy resulting from overindulgence at the evening's supper. It is, rather, a solution to Laplace's equation,

$$\nabla^2 U = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} = 0 \tag{92}$$

in spherical coördinates. This topic is treated in excruciating detail in Chapter II of the book

"The Theory of Spherical and Ellipsoidal Harmonics"
E. W. Hobson, Sc.D., LL.D., F.R.S.
Cambridge, at the University Press
1931

In essence, Eq. 92 is expressed in spherical coördinates, and a solution to the resultant partial differential equation is assumed. This assumed form leads to several ordinary differential equations which must be solved. One of the differential equations is a function of r only. The other, known as "Legendre's Equation", includes, as part of its solution, the "Legendre Polynomials and Associated Functions", $P_n^m(\sin(\phi))$. The C_{nm} and S_{nm} terms are, essentially, the constants of integration of that solution. They are determined experimentally by flying satellites, observing how they move, and choosing[10] the coefficients so the *predicted* orbits match the *observed* ones as closely as possible.

E.2.2 Gradient of $U(r, \phi, \lambda)$

Earlier, it was stated that analytical expressions for the gradient of $U(r, \phi, \lambda)$ are coded into CLOP. These expressions are given here, purely for completeness of the presentation. In an attempt to keep things under control, two simplifying symbols are defined. Let

$$\Gamma_1(m, \lambda) \stackrel{def}{=} \{C_{nm} \cos(m\lambda) + S_{nm} \sin(m\lambda)\} \tag{93}$$

and

$$\Gamma_2(m, \lambda) \stackrel{def}{=} \{C_{nm} \sin(m\lambda) - S_{nm} \cos(m\lambda)\} \tag{94}$$

[10] Through a highly sophisticated Least Squares process.

Then

$$\frac{\partial U}{\partial r} = \frac{-\mu}{r^2}\left[1 + \sum_{n=2}^{N}(n-1)\left(\frac{r_e}{r}\right)^n \sum_{m=0}^{n} P_n^m(s\phi)\Gamma_1(m,\lambda)\right] \quad (95)$$

$$\frac{1}{r\cos(\phi)}\frac{\partial U}{\partial \lambda} = \frac{-\mu}{r^2\cos(\phi)}\left[\sum_{n=2}^{N}\left(\frac{r_e}{r}\right)^n \sum_{m=0}^{n} m P_n^m(s\phi)\Gamma_2(m,\lambda)\right] \quad (96)$$

$$\frac{1}{r}\frac{\partial U}{\partial \phi} = \frac{\mu\cos(\phi)}{r^2}\left[\sum_{n=2}^{N}\left(\frac{r_e}{r}\right)^n \sum_{m=0}^{n} \frac{dP_n^m}{d[\sin(\phi)]}\Gamma_1(m,\lambda)\right] \quad (97)$$

E.2.3 Legendre Polynomials

The first few Legendre Polynomials and Associated Functions will be presented here so the reader may gain a feel for what these creatures look like. Then, the recursion formulae CLOP uses to evaluate these functions will be given. The symbol z will be used to denote the sine of the North Latitude of the radius vector to the vehicle.

$$P_0^0(z) = 1 \quad (98)$$

$$P_1^0(z) = z \quad (99)$$

$$P_2^0(z) = \frac{1}{2}(3z^2 - 1) \quad (100)$$

$$P_3^0(z) = \frac{z}{2}(5z^2 - 3) \quad (101)$$

$$P_4^0(z) = \frac{1}{8}(35z^4 - 30z^2 + 3) \quad (102)$$

$$P_1^1(z) = (1 - z^2)^{\frac{1}{2}} \quad (103)$$

$$P_2^1(z) = 3z(1 - z^2)^{\frac{1}{2}} \quad (104)$$

$$P_2^2(z) = 3(1-z^2) \tag{105}$$

$$P_3^1(z) = \frac{3}{2}(5z^2-1)(1-z^2)^{\frac{1}{2}} \tag{106}$$

$$P_3^2(z) = 15z(1-z^2) \tag{107}$$

$$P_3^3(z) = 15(1-z^2)^{\frac{3}{2}} \tag{108}$$

$$P_4^1(z) = \frac{5}{2}z(7z^2-3)(1-z^2)^{\frac{1}{2}} \tag{109}$$

$$P_4^2(z) = \frac{15}{2}(1-z^2)(7z^2-1) \tag{110}$$

$$P_4^3(z) = 105z(1-z^2)^{\frac{3}{2}} \tag{111}$$

$$P_4^4(z) = 105(1-z^2)^2 \tag{112}$$

Many recursion formulae exist for the rapid evaluation of these functions and their derivatives. Those used by CLOP depend on the value of m, and apply for those cases in which $n \geq 2$. Here they are.

$m = 0$

$$P_n^0 = ((2n-1)zP_{n-1}^0 - (n-1)P_{n-2}^0)/n \tag{113}$$

$$\frac{dP_n^0}{dz} = z\frac{dP_{n-1}^0}{dz} - nP_{n-1}^0 \tag{114}$$

$m = n$

$$P_m^m = (2m-1)yP_{m-1}^{m-1} \tag{115}$$

$$\frac{dP_m^m}{dz} = (2m-1)\left(y\frac{dP_{m-1}^{m-1}}{dz} - \frac{z}{y}P_{m-1}^{m-1}\right) \tag{116}$$

Otherwise

$$P_n^m = \left((2n-1)zP_{n-1}^m - (n+m-1)P_{n-2}^m\right)/(n-m) \quad (117)$$

$$\frac{dP_n^m}{dz} = \left((n+m)P_{n-1}^m - nzP_n^m\right)/y^2 \quad (118)$$

where

$$y = \sqrt{1-z^2} = \cos(\phi) \quad (119)$$

F The Problem With Keplerian Representation of Orbits

In Sec. 2.3.1, the state vector, **u**, was described as consisting of the normal Keplerian orbit elements; see Eq. 1 on page 12. Keplerian orbit elements assume the Earth is a perfect sphere, and assume point-masses with their associated uniform μ/r potentials. The Keplerian elements serve beautifully to describe the motion of orbiting objects, and the quantities are fairly easy to compute.

As a first attempt at modeling a non-spherical Earth, the general gravitational potential, Eq. 90 page 47, is simplified by setting $N = 2$, and keeping only the C_{20} ($-J_2$, equivalently) term. But, to solve this equation, numerous simplifying assumptions must be made to obtain any kind of solution at all. The result is the "First Order Theory" (FOT) described in Eqs. 2, 3, and 4. The implication is that these drift rates are constant. They aren't, especially in the case of $d\omega/dt$. The theory also predicts that the semi-major axis, a, the eccentricity, e, and the orbit's inclination, i, will be constant. This is *almost* true. It is *close* to true if we interpret the results as effects which apply over a single orbit, which, after all, are the conditions under which they were derived. The problem arises for an AutoNav system because they do not describe the *instantaneous* motion well at all, especially for near-circular orbits.

The FOT *is* useful for gaining broad-brush insights into orbital motion, but not for accurately predicting satellite positions. This can be seen in Fig. 18, page 65, which shows how a vehicle flying according to FOT predictions departs from motion in 16-th order gravitational field with no air-drag or other applied forces. It is seen that, for this particular circumstance, a position error of 17 km builds after only one orbit, and things get worse from there.

G Air Force Bases in the U.S.

The following is a list of location coördinates for Air Force bases located in the political United States. The original information was obtained through a literature search by the BASG technical library. The list of base coördinates is given here, followed by the AWK program which produced it from the original form.

```
!                Air Force Bases in the Political United States
!
!     Lat        Lon        Height          Location
!    (deg)      (deg)       (km)

    35.950     270.050     0.077      ! BLYTHEVILLE              AK
    34.750     267.717     0.094      ! LITTLE*ROCK              AK
    32.400     273.717     0.067      ! GUNTER                   AL
    32.367     273.700     0.051      ! MAXWELL                  AL
    64.633     212.900     0.163      ! EIELSON                  AS
    61.250     210.183     0.036      ! ELMENDORF                AS
    32.183     249.117     0.799      ! DAVIS-MONTHAN            AZ
    33.550     247.817     0.332      ! LUKE                     AZ
    33.350     248.167     0.422      ! WILLIAMS                 AZ
    39.117     238.633     0.034      ! BEALE                    CA
    37.383     239.433     0.057      ! C' TLE                   CA
    34.900     242.133     0.702      ! EI ARDS                  CA
    34.583     242.633     0.876      ! GEORGE                   CA
    34.067     241.750     0.029      ! LOS*ANGLES               CA
    33.900     242.750     0.466      ! MARCH                    CA
    38.550     238.717     0.029      ! MATHER                   CA
    34.100     242.750     0.352      ! NORTON                   CA
    38.267     238.083     0.019      ! TRAVIS                   CA
    34.683     239.517     0.122      ! VANDENBERG               CA
    38.933     255.383     1.910      ! FALCON                   CO
    39.717     255.117     1.646      ! LOWRY                    CO
    38.817     255.283     1.890      ! PETERSON                 CO
    38.983     255.133     2.219      ! US*AIR*FORCE*ACADEMY     CO
    39.117     284.517     0.009      ! DOVER                    DE
    30.483     273.500     0.702      ! EGLIN                    FL
    25.500     279.600     0.002      ! HOMESTEAD                FL
    30.417     273.317     0.011      ! HULBURT                  FL
    27.850     277.500     0.002      ! MACDILL                  FL
    28.233     279.400     0.003      ! PATRICK                  FL
    30.150     274.350     0.005      ! TYNDALL                  FL
```

30.850	276.750	0.071	! MOODY	GA
21.333	202.100	0.000	! HICKAM	HA
21.500	201.950	0.258	! WHEELER	HA
43.050	244.133	0.914	! MOUNTAIN*HOME	ID
40.300	271.850	0.224	! CHANUTE	IL
38.533	270.133	0.138	! SCOTT	IL
37.633	262.733	0.418	! MCCONNELL	KS
32.500	266.367	0.051	! BARKSDALE	LA
31.283	267.517	0.027	! ENGLAND	LA
42.483	288.717	0.041	! HANSCOM	MA
41.650	289.450	0.040	! OTIS*ANG*BASE	MA
42.183	287.433	0.074	! WESTOVER	MA
38.800	283.133	0.086	! ANDREWS	MD
46.917	292.167	0.230	! LORING	ME
46.233	275.533	0.372	! K I*SAWYER	MI
42.617	277.183	0.178	! SELFRIDGE*ANG*BASE	MI
44.450	276.600	0.193	! WURTSMITH	MI
38.717	266.617	0.265	! WHITEMAN	MO
33.650	271.550	0.065	! COLUMBUS	MS
30.417	271.083	0.008	! KEESLER	MS
47.517	248.800	1.074	! MALMSTROM	MT
35.133	281.017	0.066	! POPE	NC
35.367	282.033	0.033	! SEYMOUR*JOHNSON	NC
47.983	262.967	0.278	! GRAND*FORKS	ND
48.417	258.667	0.508	! MINOT	ND
41.150	264.050	0.319	! OFFUTT	NE
43.067	289.217	0.031	! PEASE	NH
40.017	285.400	0.041	! MCGUIRE	NJ
34.400	256.733	1.309	! CANNON	NM
32.867	253.900	1.248	! HOLLOMAN	NM
35.033	253.383	1.631	! KIRTLAND	NM
36.200	244.917	0.570	! NELLIS	NV
43.217	284.567	0.154	! GRIFFISS	NY
44.667	286.550	0.072	! PLATTSBURGH	NY
39.817	277.050	0.227	! RICKENBACKER*ANG*BASE	OH
39.783	275.950	0.251	! WRIGHT*PATTERSON	OH
34.650	260.683	0.419	! ALTUS	OK
35.483	262.467	0.393	! TINKER	OK
36.417	262.133	0.398	! VANCE	OK
32.900	279.933	0.014	! CHARLESTON	SC
33.700	281.117	0.008	! MYRTLE*BEACH	SC
33.967	279.517	0.074	! SHAW	SC
44.167	256.900	0.975	! ELLSWORTH	SD
35.350	273.800	0.305	! ARNOLD	TN

```
   30.217    262.333      0.165       ! BERGSTROM              TX
   29.350    261.567      0.183       ! BROOKS                 TX
   32.750    262.567      0.198       ! CARSWELL               TX
   32.417    260.200      0.545       ! DYESS                  TX
   31.467    259.550      0.572       ! GOODFELLOW             TX
   29.383    261.417      0.210       ! KELLY                  TX
   29.383    261.400      0.227       ! LACKLAND               TX
   29.350    259.167      0.329       ! LAUGHLIN               TX
   33.583    259.150      1.017       ! REESE                  TX
   33.983    261.483      0.309       ! SHEPPARD               TX
   41.233    248.050      1.459       ! HILL                   UT
   37.083    283.650      0.003       ! LANGLEY                VA
   47.633    242.367      0.750       ! FAIRCHILD              WA
   47.117    237.417      0.098       ! MCCHORD                WA
   41.133    255.183      1.878       ! FRANCIS*E*WARREN       WY

! There are 89 bases in this file
!
! Minimum latitude:    21.333 degrees
! Maximum latitude:    64.633 degrees
! Minimum longitude:  201.950 degrees
! Maximum longitude:  292.167 degrees
! Minimum height:       0.000 metres
! Maximum height:    2218.944 metres
```

```
BEGIN   { latmin =  1.0e10;
          latmax = -1.0e10;
          lonmin =  1.0e10;
          lonmax = -1.0e10;
          hmin   =  1.0e10;
          hmax   = -1.0e10;

          printf("!\t\tAir Force Bases in the Political United States\n");
          printf("!\n");
          printf("!     Lat       Lon       Height\t  Location\n");
          printf("!    (deg)     (deg)       (km)\n\n");
        }

NF == 9 { latitude  =          $4 + $6 / 60.0;
          longitude = 360.0 - ($7 + $9 / 60.0);
          height    =          $3 * 3.048e-4;

          latmin = latitude  < latmin ? latitude  : latmin;
          latmax = latitude  > latmax ? latitude  : latmax;

          lonmin = longitude < lonmin ? longitude : lonmin;
          lonmax = longitude > lonmax ? longitude : lonmax;

          hmin   = height < hmin      ? height    : hmin;
          hmax   = height > hmax      ? height    : hmax;

          ++bases;

          printf("%10.3f %10.3f %10.3f\t! %-24s %s\n",    latitude,
                                                          longitude,
                                                          height,
                                                          $1, $2 );
        }

END     { printf("\n\n");
          printf("! There are %d bases in this file\n",   bases       );
          printf("!\n");
          printf("! Minimum latitude:  %8.3f degrees\n", latmin       );
          printf("! Maximum latitude:  %8.3f degrees\n", latmax       );
          printf("! Minimum longitude: %8.3f degrees\n", lonmin       );
          printf("! Maximum longitude: %8.3f degrees\n", lonmax       );
          printf("! Minimum height:    %8.3f metres\n",  hmin * 1000.0);
          printf("! Maximum height:    %8.3f metres\n",  hmax * 1000.0);
        }
```

H Symbols

a	Orbit semi-major axis
e	Orbit eccentricity
i	Orbit inclination
$\mathbf{f}(\mathbf{u}, t)$	Nonlinear differential equations of motion
$\mathbf{F}$	Linearized plant geometry
$\mathbf{g}_k$	Unit vector parallel to $\mathbf{v}_k$
$\mathbf{h}(\mathbf{u}, t)$	Nonlinear measurement function
$\mathbf{H}$	Linearized measurement geometry
$\mathbf{I}$	Identity matrix
$\mathbf{J}$	Jacobian; taken with respect to $\mathbf{u}$
J_2	Second gravitational potential zonal coefficient
$\mathbf{K}$	Kalman gain matrix
μ	Earth's gravitational parameter
ν	Orbit true anomaly
ω	Orbit argument of perigee
Ω	Orbit right ascension of ascending node
$\mathbf{P}$	State covariance matrix
$\mathbf{Q}$	Plant covariance matrix
$\mathbf{R}$	Measurement covariance matrix
r_e	Equatorial radius of the Earth
τ	Orbit period
t_0	Time at Epoch
t	Time
t_k	Time at the k-th event
$\mathbf{u}$	State vector
$\mathbf{u}^k$	State vector: Keplerian elements
$\mathbf{u}^c$	State vector: Cartesian elements
$\mathbf{v}_k(\mathbf{u}, t)$	Vector from vehicle to reference point k
$\mathbf{z}$	Measurement vector

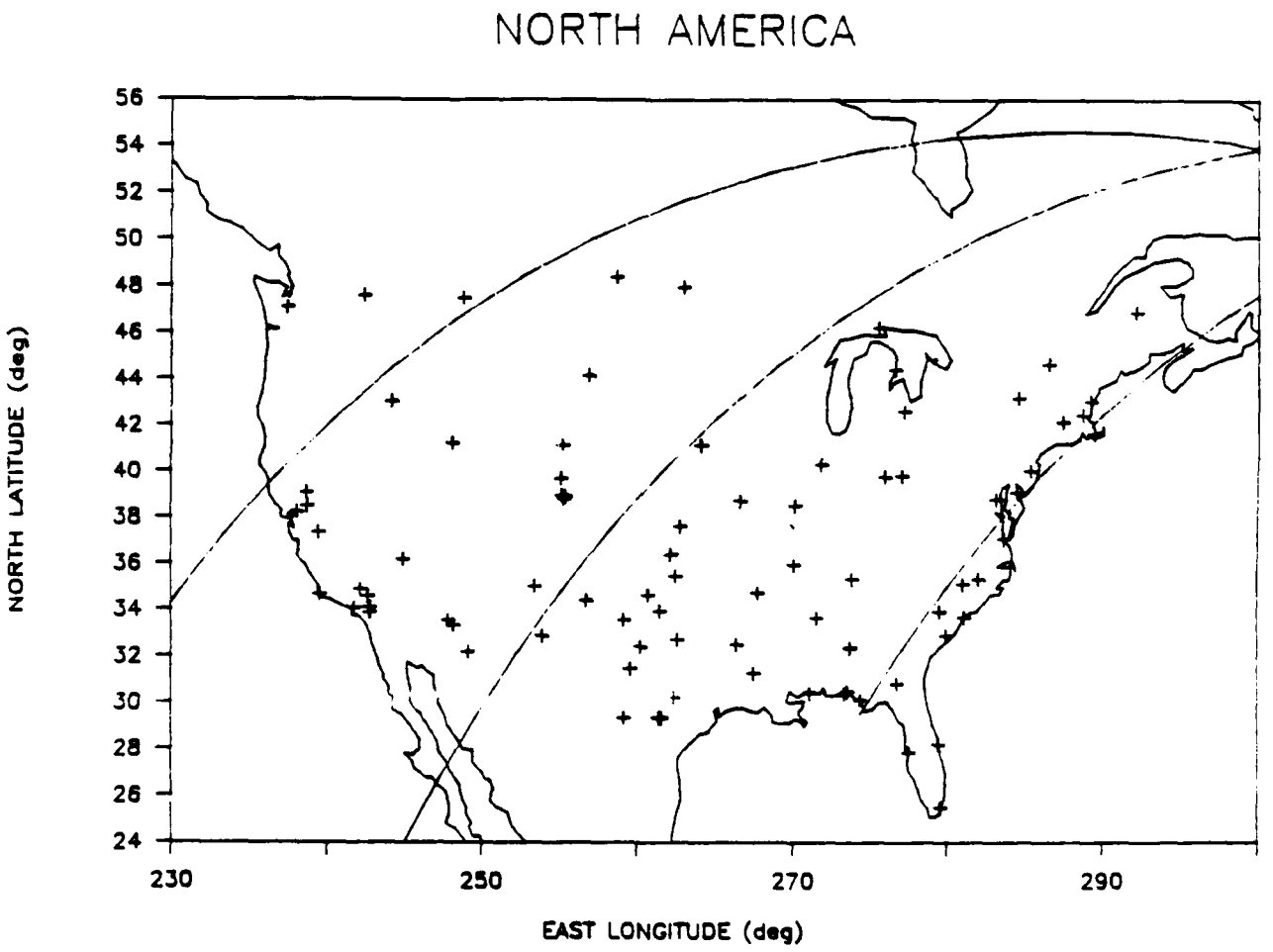

Figure 4: Ground Track of Test Orbit

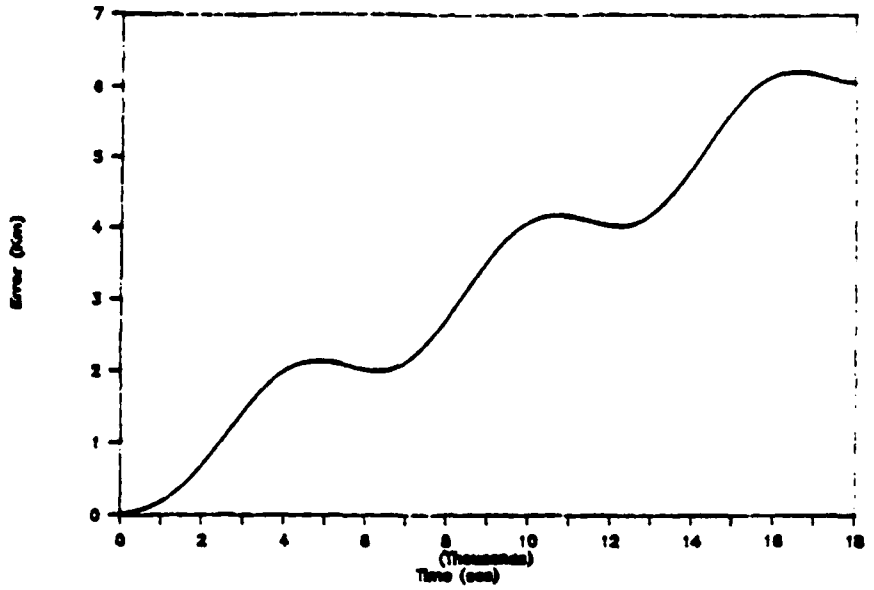

Figure 5: Effect of Small $\hat{u}$ Error

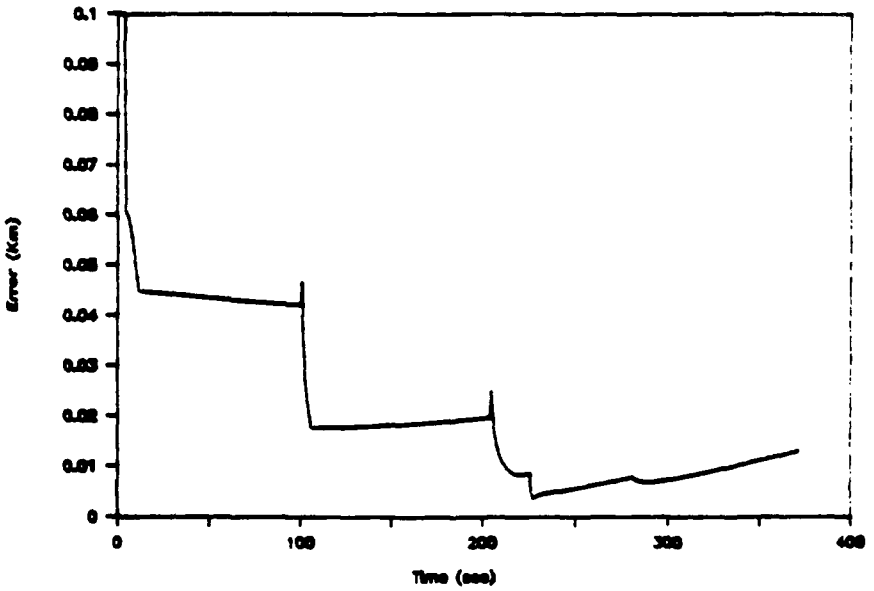

Figure 6: Flight 1

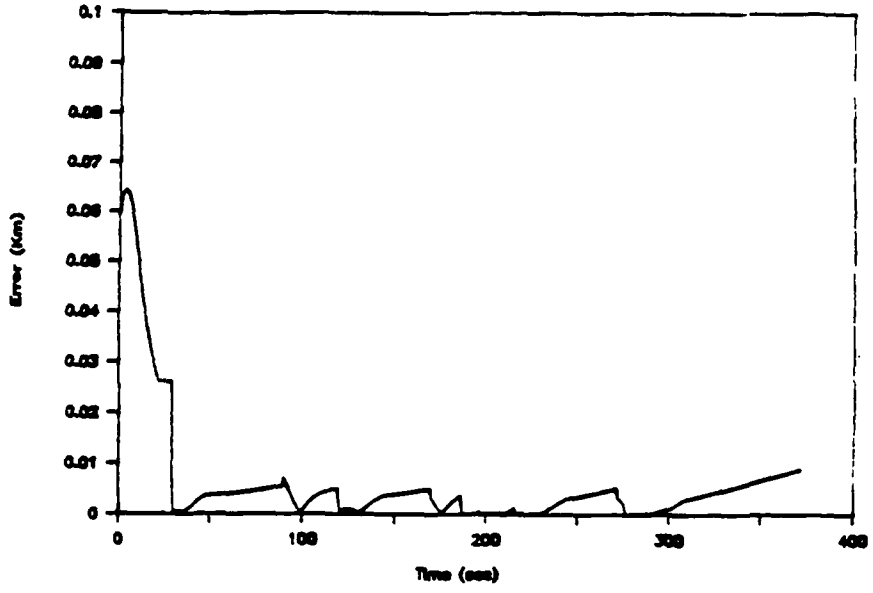

Figure 7: Flight 1a

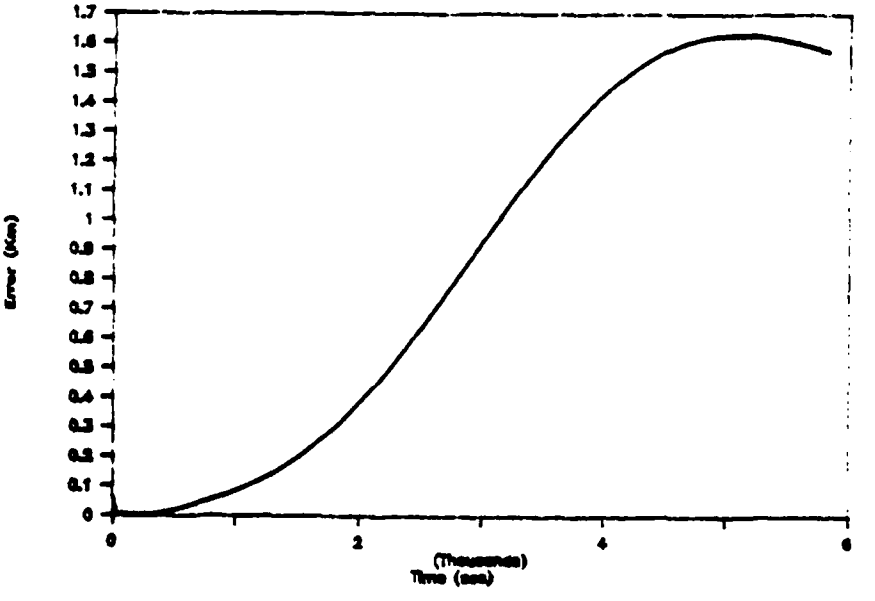

Figure 8: Flight 1b

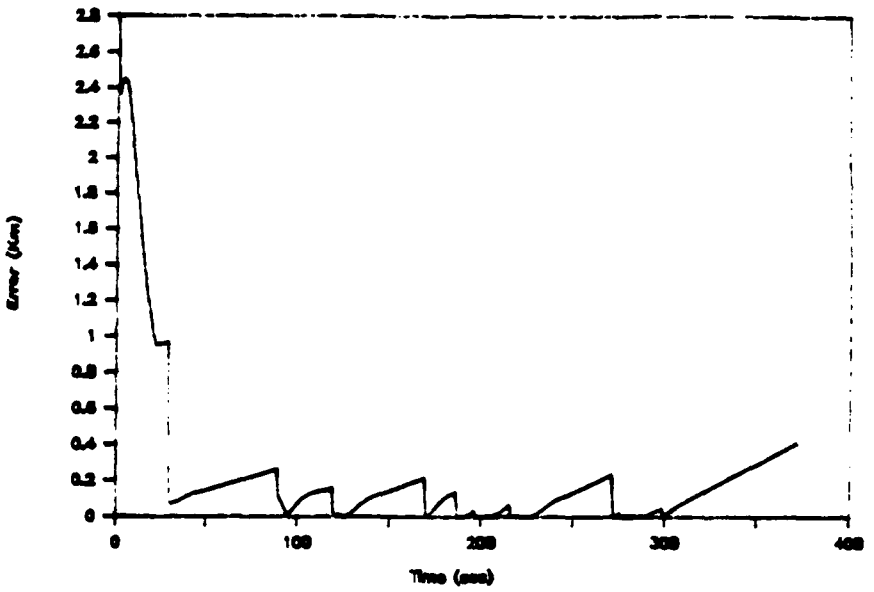

Figure 9: Flight 1c

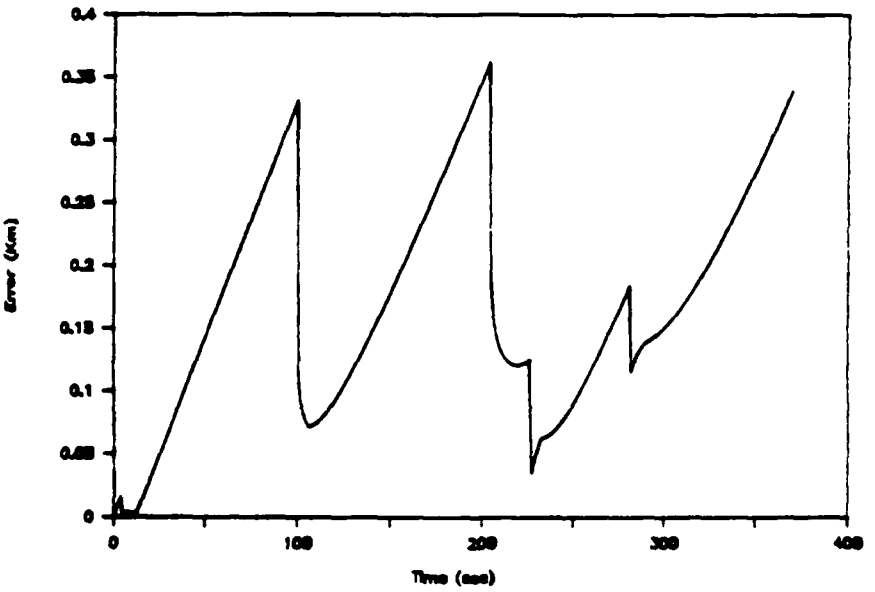

Figure 10: Flight 2

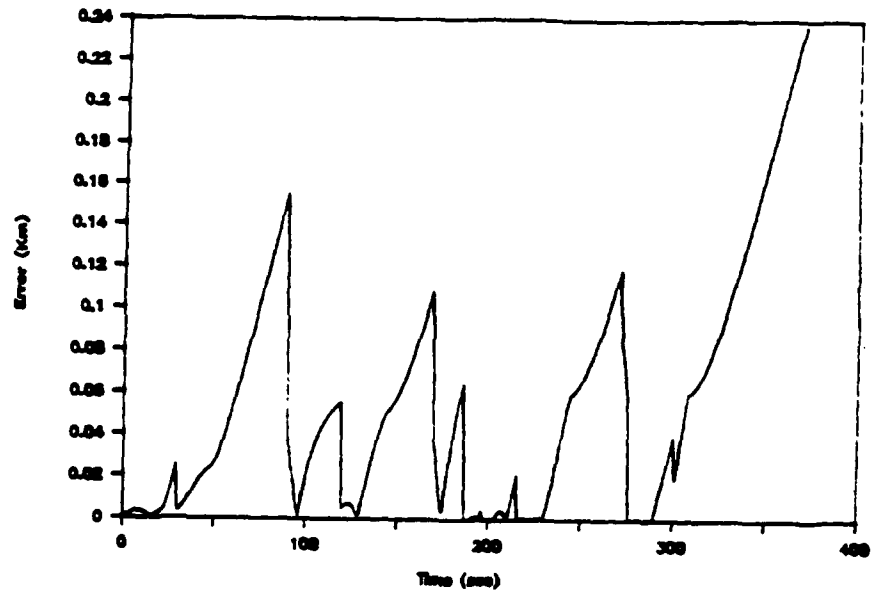

Figure 11: Flight 2a

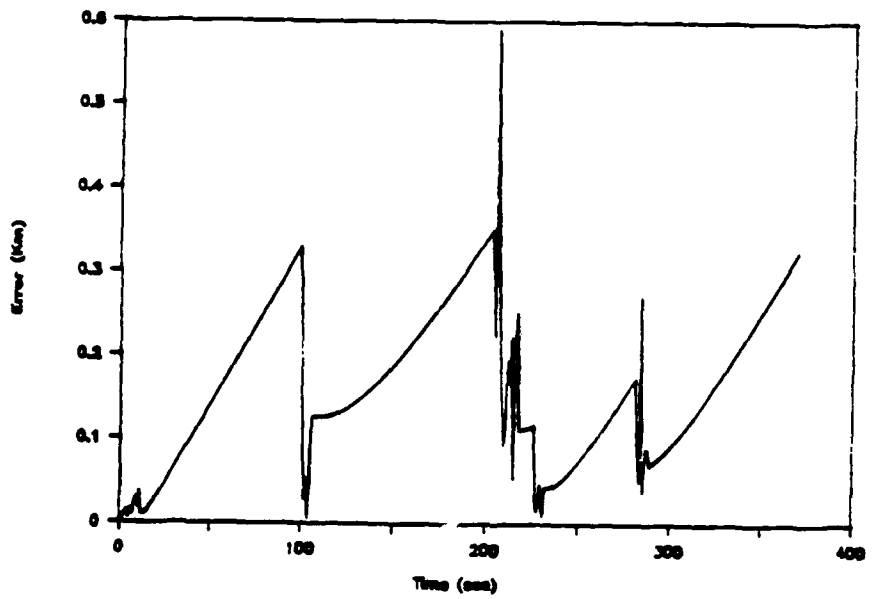

Figure 12: Flight 3

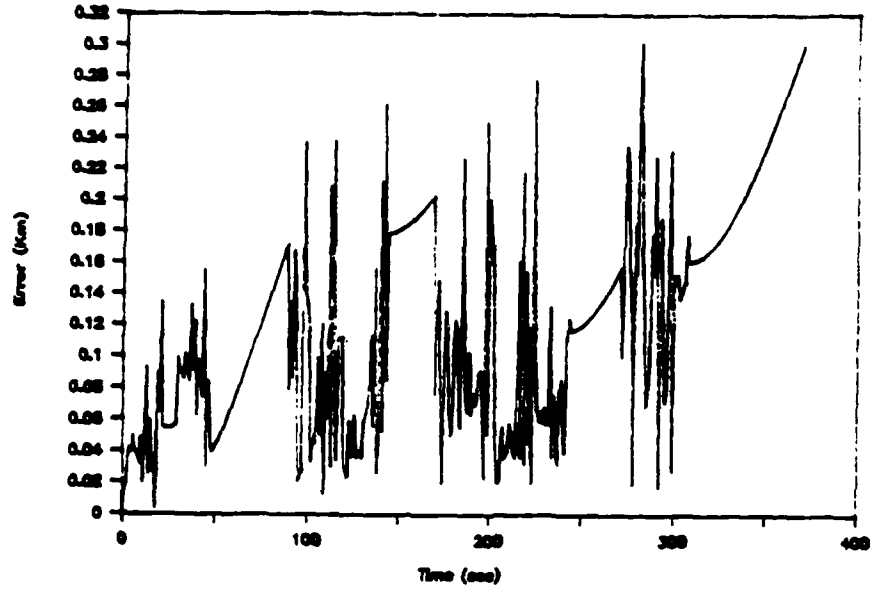

Figure 13: Flight 3a

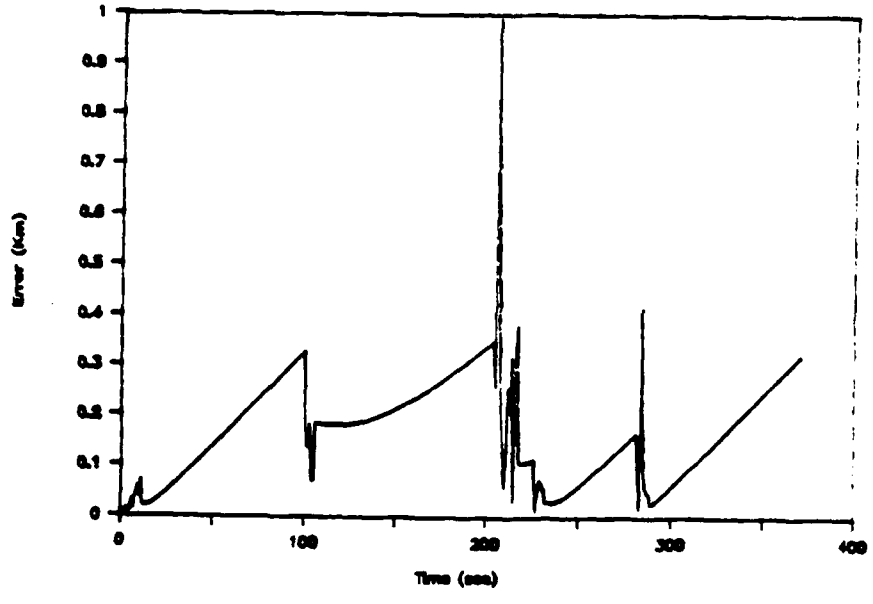

Figure 14: Flight 3b

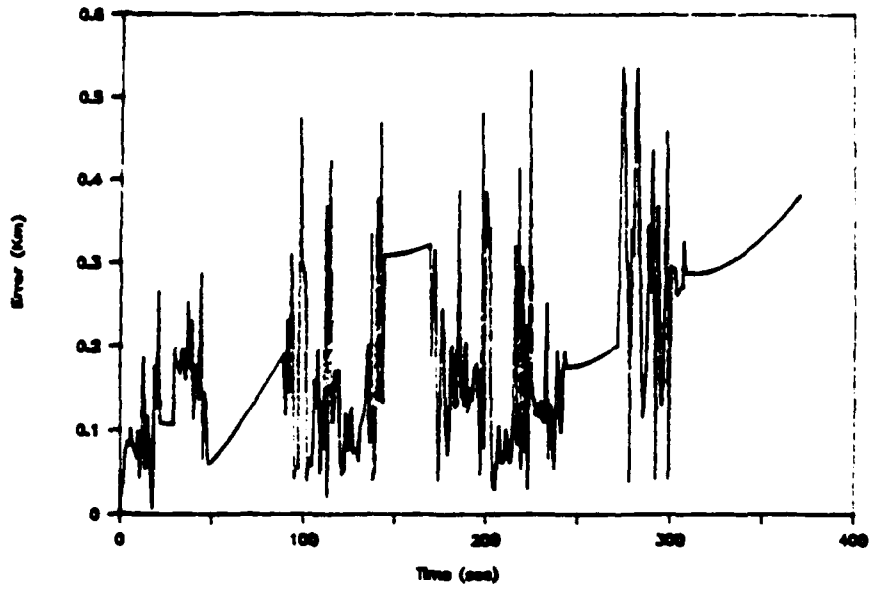

Figure 15: Flight 3c

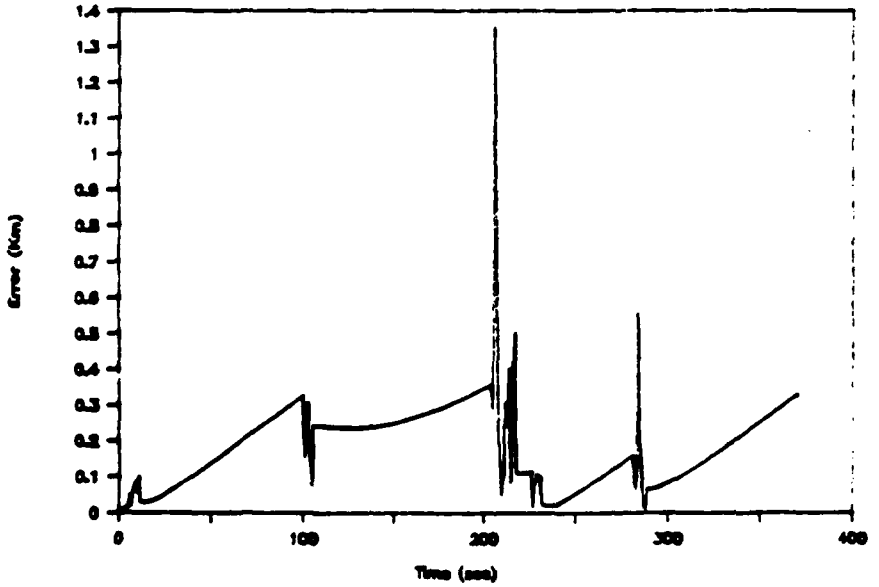

Figure 16: Flight 3d

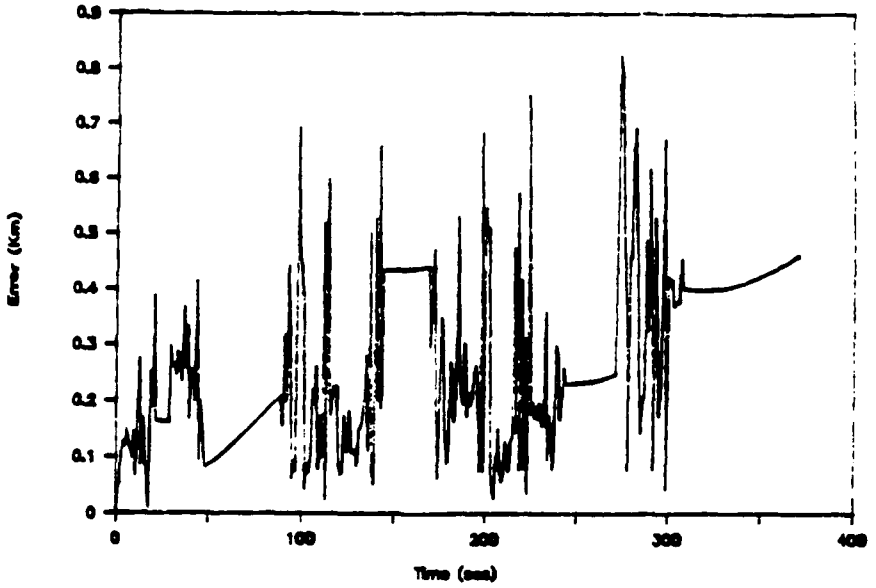

Figure 17: Flight 3e

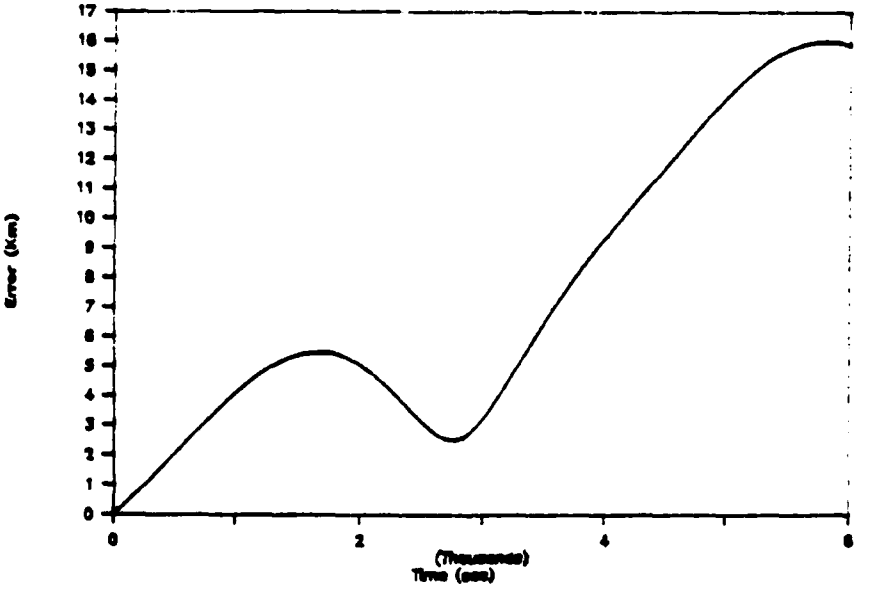

Figure 18: Keplerian Representation Errors